รหัสโครงการ SUT7-705-48-24-13

รายงานการวิจัย

การประมวลผลหลังกระบวนการทำเหมืองข้อมูล

(Post- data mining processing)

ผู้วิจัย

หัวหน้าโครงการ

รองศาสตราจารย์ ดร.นิตยา เกิดประสพ

สาขาวิชาวิศวกรรมคอมพิวเตอร์

สำนักวิชาวิศวกรรมศาสตร์

ได้รับทุนอุดหนุนการวิจัยจากมหาวิทยาลัยเทคโนโลยีสุรนารี ปีงบประมาณ พ.ศ. 2548 และ 2549

ผลงานวิจัยเป็นความรับผิดชอบของหัวหน้าโครงการวิจัยแต่เพียงผู้เดียว

ธันวาคม 2552

กิตติกรรมประกาศ

ผู้วิจัยขอขอบคุณมหาวิทยาลัยเทคโนโลยีสุรนารีและสำนักงานคณะกรรมการวิจัยแห่งชาติ ที่ได้จัดสรรงบประมาณในการทำวิจัยให้ในปีงบประมาณ 2548 และ 2549 โครงการนี้ยังได้รับงบประมาณบางส่วน รวมถึงความร่วมมือในการดำเนินงานจากหน่วยปฏิบัติการวิจัยด้านวิศวกรรมข้อมูลและการค้นหาความรู้ (Data Engineering and Knowledge Discovery -- DEKD -- Research Unit) ผู้วิจัยขอขอบคุณผู้ทรงคุณวุฒิที่ได้เสียสละเวลาทำหน้าที่ตรวจข้อเสนอโครงการ และตรวจร่างรายงานการวิจัยฉบับสมบูรณ์

บทคัดย่อภาษาไทย

กระบวนการทำเหมืองข้อมูลประกอบด้วยเจ็ดขั้นตอนหลักคือ (๑) การรวบรวมข้อมูล, (๒) การแปลงรูปแบบข้อมูล, (๓) การปรับปรุงข้อมูล, (๔) การคัดเลือกข้อมูล, (๕) การค้นหาเพื่อคัดแยกแพทเทิร์นของข้อมูล หรือเรียกว่าการทำเหมืองความรู้, (๖) การประเมินคุณภาพของแพทเทิร์น, และ (๗) การนำเสนอความรู้ ขั้นตอนที่ ๑ ถึง ๔ จัดเป็นขั้นตอนก่อนการทำเหมืองข้อมูล ในขณะที่ขั้นตอนที่ ๖ และ ๗ อาจจัดได้ว่าเป็นขั้นตอนหลังการทำเหมืองข้อมูล ดังนั้นทั้งเจ็ดขั้นตอนสามารถจัดรวมเป็นขั้นก่อนการทำเหมืองข้อมูล ขณะทำเหมืองข้อมูล และหลังการทำเหมืองข้อมูล งานวิจัยนี้เน้นการศึกษาที่ขั้นหลังการทำเหมืองข้อมูล ระบบทำเหมืองข้อมูลส่วนมากจะสิ้นสุดกระบวนการที่ขั้นตอนนำเสนอความรู้ แต่ในงานวิจัยนี้ได้นำเสนอขั้นตอนต่อจากนั้น คือ ขั้นตอนการใช้ประโยชน์จากความรู้ที่ได้จากการทำเหมืองข้อมูล งานวิจัยนี้แสดงวิธีการนำความรู้ในรูปแบบของกฎการจำแนกมาใช้ประโยชน์ โดยความรู้ในรูปแบบของกฎการจำแนกจะถูกนำมาประเมินคุณภาพด้วยเกณฑ์ความครอบคลุมข้อมูล กฎที่ครอบคลุมข้อมูลได้มากจะได้รับการคัดเลือกมาใช้ในระบบผู้เชี่ยวชาญ การทดสอบความสามารถของระบบผู้เชี่ยวชาญที่ใช้การสร้างฐานความรู้แบบอัตโนมัตินี้ ใช้วิธีวัดความถูกต้องของคำแนะนำที่ให้โดยระบบผู้เชี่ยวชาญ จากนั้นเปรียบเทียบกับความถูกต้องของการทำนายโดยโปรแกรมการทำเหมืองข้อมูลแบบจำแนก รวมสามโปรแกรมได้แก่ โปรแกรมที่ใช้หลักการสร้างต้นไม้ตัดสินใจ (ID3) โปรแกรมสร้างกฎการจำแนก (PRISM) และโปรแกรมข่ายงานประสาท (Neural network) ผลการทดสอบยืนยันความถูกต้องในการให้คำแนะนำของระบบผู้เชี่ยวชาญ

บทคัดย่อภาษาอังกฤษ

The process of data mining comprises of seven major steps: (1) data integration, (2) data transformation, (3) data cleaning, (4) data selection, (5) pattern extraction or knowledge mining, (6) pattern evaluation, and (7) knowledge presentation. Steps 1 to 4 are pre-data mining, whereas steps 6 and 7 may be viewed as post-data mining. Therefore, the seven major steps can be grouped into pre-data mining, mining, and post-data mining. This research studies the post-data mining processing. Most data mining systems finish their processing at the knowledge presentation step. Our work further the post-data mining processing to the step of knowledge deployment. This research illustrates the knowledge deployment step in which its input is the induced knowledge, in the formalism of classification rules. These rules are evaluated and filtered on the basis of coverage measurement. High coverage rules are transformed into decision rules to be used by the inference engine of the expert system. The accuracy of recommendation given by the expert system is evaluated and compared to other three classification systems, i.e. decision-tree induction (ID3), rule induction (PRISM), and neural network. The experimental results confirm the high accuracy of our expert system and the induced knowledge base.

สารบัญ

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

บทที่ 1
บทนำ

ความสำคัญและที่มาของปัญหาการวิจัย

ความสำคัญของการทำเหมืองข้อมูล

ในปัจจุบันเครื่องคอมพิวเตอร์และอุปกรณ์นำเข้าข้อมูล เช่น สแกนเนอร์ เครื่องอ่านบาร์โค้ด มีใช้อย่างแพร่หลาย ประกอบกับอุปกรณ์เก็บข้อมูล เช่น ฮาร์ดดิสก์ มีราคาถูกลง ทำให้ข้อมูลที่ถูกบันทึกอยู่ในรูปแบบดิจิตอลมีปริมาณมหาศาล การใช้แรงงานคนวิเคราะห์ข้อมูลเพื่อจะนำความรู้จากข้อมูลมาใช้ให้เกิดประโยชน์ทันเวลา เป็นสิ่งที่แทบจะเป็นไปไม่ได้

ปัญหาของปริมาณข้อมูลจะเห็นได้ชัดเจนในกรณีของข้อมูลที่ได้รับจากดาวเทียมสำรวจสภาพอากาศและพื้นผิวโลกที่ส่งข้อมูลมายังสถานีภาคพื้นดินด้วยขนาดหลายเทอราไบต์ทุกวัน และในอนาคตอันใกล้เมื่อแผ่นวงจรคอมพิวเตอร์มีขนาดเล็กมากและราคาถูกลงจนกระทั่งสามารถติดแผ่นวงจรที่สินค้าทุกชิ้น หรือที่วัตถุทุกชนิด เพื่อให้สามารถติดตามตำแหน่งและข้อมูลอื่นๆของวัตถุนั้นๆ ปริมาณข้อมูลที่รวบรวมได้จะยิ่งทวีจำนวนขึ้นมากจนกระทั่งไม่สามารถใช้ผู้เชี่ยวชาญมาวิเคราะห์ข้อมูลเหล่านั้นเพื่อให้สามารถนำความรู้มาใช้ประโยชน์ได้ทันต่อความต้องการ

แนวทางที่จะช่วยแก้ไขปัญหาความล่าช้าของการวิเคราะห์ข้อมูล ทำได้โดยการปรับปรุงกระบวนการวิเคราะห์ข้อมูลให้เป็นอัตโนมัติมากขึ้น ลดขั้นตอนการควบคุมและสั่งงานจากผู้เชี่ยวชาญให้น้อยลง โดยให้ระบบคอมพิวเตอร์ทำหน้าที่ค้นหาแนวโน้มและรูปแบบต่างๆที่น่าสนใจจากข้อมูล หรือวิเคราะห์ความสัมพันธ์ภายในกลุ่มข้อมูลได้ด้วยความสามารถของระบบเอง กระบวนการวิเคราะห์และค้นหาความรู้จากข้อมูลโดยอัตโนมัตินี้เรียกว่า **การทำเหมืองข้อมูล** (data mining) ผลลัพธ์ที่ได้จากการทำเหมืองข้อมูล คือ **ความรู้** (knowledge) ซึ่งอาจจะเป็นได้หลายรูปแบบ เช่น โมเดลหรือแพทเทิร์นที่อธิบายลักษณะของข้อมูลส่วนใหญ่, กฎที่จะใช้ช่วยทำนายข้อมูลหรือเหตุการณ์ที่จะเกิดขึ้นในอนาคต, ความสัมพันธ์ระหว่างข้อมูล

การใช้ประโยชน์เทคโนโลยีการทำเหมืองข้อมูล

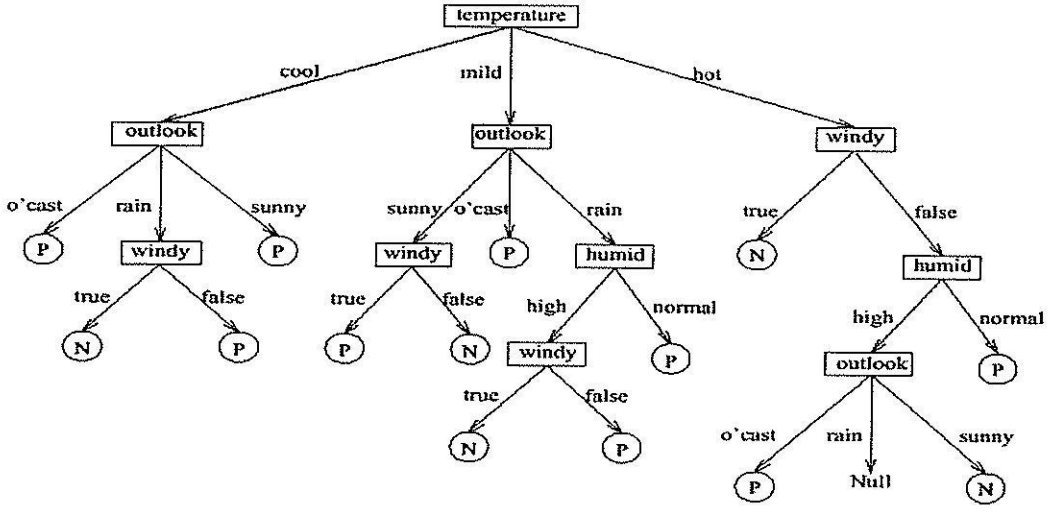
การใช้ประโยชน์เทคโนโลยีการทำเหมืองข้อมูลที่เป็นที่นิยม โดยเฉพาะอย่างยิ่งทางภาคธุรกิจจะกระทำในสองลักษณะคือใช้เพื่อการจำแนกหาลักษณะข้อมูลในแต่ละกลุ่ม (classification) และใช้เพื่อค้นหาความสัมพันธ์ภายในกลุ่มข้อมูล (association)

ประโยชน์ที่ได้จากการจำแนกข้อมูล คือใช้แบบแผนหรือแพทเทิร์นที่ปรากฏในข้อมูลเป็นเครื่องมือในการคาดหมายข้อมูลในอนาคตหรือใช้ประกอบการตัดสินใจ ตัวอย่างเช่นในตาราง

ที่ 1.1 (Quinlan, 1992) เป็นข้อมูลสภาพอากาศที่รวบรวมมาภายใน 14 วัน แอททริบิวต์สุดท้ายของข้อมูล (Class) แสดงการตัดสินใจของนักกอล์ฟว่าจากสภาพอากาศในแต่ละวันนักกอล์ฟตัดสินใจออกไปเล่นกอล์ฟ (P = Play) หรืองดเล่นกอล์ฟในวันนั้นๆ (N = don't play) และผลลัพธ์ของการทำเหมืองข้อมูลประเภทเพื่อการจำแนก (classification) แสดงได้ในลักษณะของต้นไม้ตัดสินใจ ดังรูปที่ 1.1

ตารางที่ 1.1 ข้อมูลสภาพอากาศที่ใช้ประกอบการตัดสินใจเล่นกอล์ฟ

No.	Attributes				Class
	Outlook	Temperature	Humidity	Windy	
1	sunny	hot	high	false	N
2	sunny	hot	high	true	N
3	overcast	hot	high	false	P
4	rain	mild	high	false	P
5	rain	cool	normal	false	P
6	rain	cool	normal	true	N
7	overcast	cool	normal	true	P
8	sunny	mild	high	false	N
9	sunny	cool	normal	false	P
10	rain	mild	normal	false	P
11	sunny	mild	normal	true	P
12	overcast	mild	high	true	P
13	overcast	hot	normal	false	P
14	rain	mild	high	true	N



รูปที่ 1.1 โครงสร้างต้นไม้ตัดสินใจที่อธิบายเงื่อนไขการตัดสินใจเล่นกอล์ฟ

ต้นไม้ตัดสินใจนี้คือรูปแบบหนึ่งของความรู้ที่เป็นผลลัพธ์ของการทำเหมืองข้อมูลแบบจำแนก สามารถนำไปใช้ประโยชน์ในการคาดหมายเหตุการณ์ในอนาคต เช่น ถ้าในสัปดาห์ต่อไปพยากรณ์อากาศระบุว่า อากาศร้อน (temperature = hot) และลมพัดแรง (windy = true) ทำนายได้ว่านักกอล์ฟจะไม่ออกไปเล่นกอล์ฟ

การทำเหมืองข้อมูลเพื่อค้นหาความสัมพันธ์ภายในกลุ่มข้อมูล (association) มักจะใช้ประโยชน์เพื่อการวางแผนเพื่อบริหารจัดการ เช่น ข้อมูลการซื้อสินค้าของลูกค้าซุปเปอร์มาเก็ตในรอบหนึ่งเดือนที่ผ่านมา เมื่อวิเคราะห์หาความสัมพันธ์แล้วพบว่า*"ถ้าลูกค้าซื้อนมและขนมปังแล้ว ลูกค้าจะซื้อบะหมี่สำเร็จรูปด้วย"* ผลลัพธ์ที่ได้จากการทำเหมืองข้อมูลตามตัวอย่างนี้ คือ ความรู้เกี่ยวกับพฤติกรรมของผู้บริโภค ที่จะช่วยให้ผู้จัดการซุปเปอร์มาเก็ตสามารถวางแผนการจัดวางสินค้า และการจัดโปรแกรมกระตุ้นยอดขายสินค้าบางรายการได้อย่างมีประสิทธิภาพ

อุปสรรคของการใช้ประโยชน์เทคโนโลยีการทำเหมืองข้อมูล

การทำเหมืองข้อมูลเป็นเทคโนโลยีใหม่ที่มีจุดมุ่งหมายให้การวิเคราะห์ข้อมูล และการใช้ประโยชน์จากข้อมูลเป็นไปได้อย่างรวดเร็วและสะดวกมากขึ้น แต่อุปสรรคที่สำคัญคือความรู้ที่ค้นพบโดยโปรแกรมทำเหมืองข้อมูล มีจำนวนมาก แต่ความรู้ที่เป็นประโยชน์โดยแท้จริงมีน้อยมาก ตัวอย่างเช่นข้อมูลสภาพอากาศและการตัดสินใจเล่นกอล์ฟในตารางที่ 1.1 ที่มีเพียง 14 รายการ เมื่อทำเหมืองข้อมูลแบบหาความสัมพันธ์ (association) จะได้ความรู้ในรูปของกฎ "ถ้าเกิดเหตุการณ์ x แล้ว จะเกิดเหตุการณ์ y" หรือ x ==> y เป็นจำนวนมากถึง 58 กฎ ดังแสดงรายละเอียดในรูปที่ 1.2

```
Apriori
Minimum support: 0.05
Minimum metric <confidence>: 0.9
Best rules found:
 1. humidity=normal windy=FALSE 4 ==> play=yes 4    conf:(1)
 2. temperature=cool 4 ==> humidity=normal 4    conf:(1)
 3. outlook=overcast 4 ==> play=yes 4    conf:(1)
 4. temperature=cool play=yes 3 ==> humidity=normal 3    conf:(1)
 5. outlook=rainy windy=FALSE 3 ==> play=yes 3    conf:(1)
 6. outlook=rainy play=yes 3 ==> windy=FALSE 3    conf:(1)
...          ...
53. windy=FALSE play=no 2 ==> outlook=sunny 2    conf:(1)
54. outlook=sunny humidity=normal 2 ==> play=yes 2    conf:(1)
55. outlook=sunny play=yes 2 ==> humidity=normal 2    conf:(1)
56. outlook=sunny temperature=hot 2 ==> play=no 2    conf:(1)
57. temperature=hot play=no 2 ==> outlook=sunny 2    conf:(1)
58. outlook=sunny temperature=hot 2 ==> humidity=high 2    conf:(1)
```

รูปที่ 1.2 ความรู้ในลักษณะของกฎอธิบายความสัมพันธ์

จากตัวอย่างข้างต้นข้อมูลมีเพียง 14 รายการ แต่เมื่อทำเหมืองข้อมูลแล้วได้ความรู้ปริมาณมากถึง 58 รายการ ความรู้ที่ได้จำนวนมากนี้มีเพียงบางรายการเท่านั้นที่จะนำไปใช้ช่วยในการวางแผนตัดสินใจได้ ในการใช้งานเทคโนโลยีการทำเหมืองข้อมูลกับข้อมูลที่เกิดขึ้นจริง ปริมาณข้อมูลตั้งต้นจะไม่ใช่เพียง 14 รายการ แต่จะมากเป็นล้านล้านรายการ ซึ่งปริมาณความรู้ที่เป็น

ผลลัพธ์ของการทำเหมืองข้อมูลก็จะมากขึ้นเช่นเดียวกัน นอกจากนี้ในบางครั้งความรู้ที่ค้นพบได้มีความซ้ำซ้อนกัน หรือในบางกรณีเป็นความรู้ที่ไม่ใช่การค้นพบใหม่ เช่น "พ่อบ้านในทุกครัวเรือนเป็นเพศชาย" ซึ่งถึงแม้ว่าจะเป็นความรู้ที่ถูกต้องเที่ยงตรง แต่ไม่เกิดประโยชน์

ดังนั้นขั้นตอนสุดท้ายที่สำคัญของการใช้เทคโนโลยีการทำเหมืองข้อมูลเพื่อการวิเคราะห์ข้อมูลอัตโนมัติ คือ การคัดเลือกและกลั่นกรองความรู้ที่ได้ว่าจะนำความรู้ใดรายงานต่อผู้บริหารเพื่อใช้ประโยชน์ในการตัดสินใจ และความรู้ใดควรจะทิ้งไปเพราะซ้ำซ้อน หรือไม่เกิดประโยชน์ ในปัจจุบันกระบวนการพิจารณาและคัดแยกความรู้นี้กระทำโดยแรงงานผู้เชี่ยวชาญ ที่มีความเข้าใจลักษณะของข้อมูล และมีความสามารถในการตีความผลลัพธ์ที่ได้จากการทำเหมืองข้อมูล ในหน่วยงานหรือองค์กรขนาดใหญ่ จึงต้องใช้ทีมบุคลากรที่มีประสบการณ์ด้านการทำเหมืองข้อมูลจำนวนมาก การจะพัฒนาเทคโนโลยีการทำเหมืองข้อมูลให้หน่วยงานขนาดเล็กหรือบุคคลทั่วไปใช้ประโยชน์ได้อย่างทั่วถึง จำเป็นจะต้องมีการพัฒนากระบวนการคัดเลือกความรู้รวมถึงรวบรวมฐานความรู้ให้เป็นอัตโนมัติมากขึ้น ใช้แรงงานผู้เชี่ยวชาญให้น้อยลงตามลำดับ

การพิจารณาความน่าสนใจ (interestingness) ของความรู้ที่ค้นพบโดยการทำเหมืองข้อมูลเป็นประเด็นปัญหาสำคัญ ที่ได้รับความสนใจจากนักวิจัยจำนวนมากตั้งแต่ยุคเริ่มต้นของงานวิจัยด้านการทำเหมืองข้อมูล (Piatetsky-Shapiro and Matheus 1994; Piatetsky-Shapiro et al. 1994) เนื่องจากได้เห็นถึงความสำคัญว่าความรู้ที่จะนำไปใช้ประโยชน์ได้ จะต้องเป็นความรู้ที่เป็นการค้นพบใหม่และน่าสนใจ โดย Piatetsky-Shapiro และ Matheus (1994) ได้พัฒนาระบบชื่อ KEFIR ขึ้นมาเพื่อใช้กับงานด้านการประกันสุขภาพ ระบบจะพิจารณาข้อมูลที่เบี่ยงเบนไปจากข้อมูลปกติและตัดสินว่าข้อมูลที่เบี่ยงเบนนี้เป็นข้อมูลที่ควรค่าแก่การให้ความสนใจ การพิจารณาเกณฑ์ความน่าสนใจโดยระบบนี้จึงเป็นการพิจารณาในขอบเขตที่จำกัดมาก ในช่วงปีต่อมา Silberschatz และ Tuzhilin (1995; 1996) ได้ร่วมมือกันเสนอวิธีการพิจารณาความน่าสนใจของความรู้ที่ค้นพบโดยใช้ probabilistic belief เป็นแนวทางในการพิจารณาความน่าจะเป็นว่าความรู้ใดที่ผู้ใช้จะเห็นว่าน่าสนใจ

ปัจจัยที่นักวิจัยใช้พิจารณาประเด็นความน่าสนใจมีได้หลายปัจจัย ได้แก่ จำนวนข้อมูลที่ความรู้นั้นสามารถอธิบายได้ (coverage) ความถูกต้องของความรู้ (confidence) ความไม่แปรผันของความรู้ (strength) ความสำคัญของความรู้ (significance) ความเข้าใจง่ายของความรู้ (simplicity) ความใหม่ของการค้นพบ (unexpectedness) และเป็นความรู้ที่นำไปสู่การปฏิบัติได้ (actionability) ปัจจัยเหล่านี้ถูกหยิบยกขึ้นมาพิจารณาโดยนักวิจัยหลายคณะ (Major and Mangano 1993; Piatetsky-Shapiro and Matheus 1994; Quinlan 1992)

ปัจจัยในด้าน coverage, confidence, strength, significance, simplicity เป็นปัจจัยที่สามารถสร้างเทคนิคขึ้นมาจัดการได้ โดยไม่ต้องใช้ข้อมูลหรือคำแนะนำประกอบจากผู้ใช้หรือจาก

ความรู้อื่นที่เกี่ยวเนื่องกัน แต่ปัจจัยด้าน unexpectedness และ actionability เป็นปัจจัยที่พิจารณาตัดสินได้ยาก

ในส่วนของการจัดการกับปริมาณความรู้ ที่เป็นผลผลิตของกระบวนการทำเหมืองข้อมูล และมักจะได้ความรู้ที่ไม่เป็นประโยชน์ปะปนออกมาเป็นจำนวนมาก เป็นปัญหาที่มีผลกระทบอย่างมากกับงานทำเหมืองข้อมูลโดยเฉพาะงานประเภท association (Bayardo et al. 1999; Brin et al. 1991; Frawley et al. 1991; Klemettinen et al. 1994; Suzuki 1997) นักวิจัยส่วนใหญ่จะใช้วิธีจัดการระหว่างกระบวนการทำเหมืองข้อมูล เช่นใช้วิธีตัดทิ้ง (pruning) เส้นทางค้นหาที่คาดว่าจะนำไปสู่ความรู้ที่ไม่เกิดประโยชน์ นักวิจัยในกลุ่มนี้ได้แก่ Quinlan (1992), Breiman และคณะ (1984), Clark และ Matwin (1993) นักวิจัยในอีกกลุ่มจะใช้วิธีนำความรู้อื่น (domain knowledge) มาใช้ช่วยในระหว่างการค้นหาความรู้ โดยใช้สมมุติฐานว่าความรู้อื่นๆที่เกี่ยวข้องจะช่วยในการตัดสินใจขจัดทิ้งผลลัพธ์ของการทำเหมืองข้อมูลที่จะไม่ก่อประโยชน์กับงาน นักวิจัยในกลุ่มนี้ได้แก่ Ortega และ Fisher (1995), Clark และ Matwin (1993), Pazzani และ Kibler (1992)

ถึงแม้แนวทางของเทคนิค pruning และเทคนิคการใช้ domain knowledge จะช่วยลดปริมาณความรู้ที่ไม่ก่อประโยชน์ แต่วัตถุประสงค์ของนักวิจัยในทั้งสองกลุ่มนี้มุ่งเน้นไปที่การลดปริมาณความรู้เพื่อเพิ่มประสิทธิภาพของกระบวนการค้นหาความรู้ (induction process) มากกว่าจะเพื่อเอื้อประโยชน์ให้กับผู้ใช้ ให้สามารถใช้ประโยชน์จากผลลัพธ์ของการทำเหมืองข้อมูลได้อย่างเต็มประสิทธิภาพและเกิดประโยชน์ในเวลาอันรวดเร็ว

ในระยะหลังของงานวิจัยในสาขาการทำเหมืองข้อมูล นักวิจัยเริ่มตระหนักมากขึ้นว่ากระบวนการจัดการกับผลลัพธ์(หรือ ความรู้) ที่ได้หลังจากการทำเหมืองข้อมูลเป็นสิ่งจำเป็นเพื่อให้การใช้ประโยชน์จากการทำเหมืองข้อมูลเป็นอัตโนมัติและทันต่อความต้องการมากขึ้น ดังจะเห็นได้จากงานวิจัยของ Wang, Tay, and Liu (1998) และ Adomavicius and Tuzhilin (2001)

งานวิจัยที่พัฒนาขึ้นนี้อยู่ในแนวทางของกระบวนการจัดการกับความรู้ ซึ่งจัดเป็น post-processing ของกระบวนการทำเหมืองข้อมูล โดยมีการพัฒนาเทคนิคและกระบวนการ ในการตรวจสอบความรู้ที่เป็นผลลัพธ์จากการทำเหมืองข้อมูล ให้สามารถแยกแยะความรู้ที่เป็นประโยชน์ออกจากความรู้ที่ใช้ประโยชน์ไม่ได้ จากนั้นเลือกไว้เฉพาะความรู้ที่ผ่านการคัดสรรแล้ว ส่งต่อไปยังฐานความรู้กลางเพื่อใช้ประโยชน์ในงานด้านการสนับสนุนการตัดสินใจ (decision support) การคัดเลือกความรู้ในงานวิจัยนี้ใช้เกณฑ์ coverage เพื่อแปลงเป็นความน่าจะเป็นที่ความรู้นั้นจะสามารถประยุกต์ใช้ได้ ประกอบกับการใช้ค่า threshold ที่ผู้ใช้ระบุเป็นเกณฑ์ขั้นต่ำในการคัดเลือกความรู้ ดังนั้นในงานวิจัยนี้จะใช้คำว่า probability แทนความหมายของ coverage ทั้งนี้เพื่อให้สอดคล้องกับการนำความรู้ไปใช้ประโยชน์ โดยเป็นส่วนหนึ่งของฐานความรู้ในระบบผู้เชี่ยวชาญ

เทคนิคการทำเหมืองข้อมูลเพื่อค้นหาความรู้ ในงานวิจัยนี้จะเน้นที่เทคนิคการสร้างกฎในลักษณะของ IF-THEN ทั้งนี้เพื่อให้ได้รูปแบบของความรู้ที่สอดคล้องกับรูปแบบที่ใช้ในฐานความรู้ของระบบผู้เชี่ยวชาญ กฎเหล่านี้เป็นกฎที่ได้จากการแปลงโครงสร้างต้นไม้ตัดสินใจที่ถูกสร้างในระหว่างกระบวนการทำเหมืองข้อมูล

วัตถุประสงค์ของการวิจัย

โครงการวิจัยนี้มีวัตถุประสงค์เพื่อกำหนดแนวทางของการตรวจสอบและประเมินความรู้ที่เป็นผลลัพธ์จากการทำเหมืองข้อมูล ความรู้ดังกล่าวจะถูกแสดงอยู่ในลักษณะของกฎเช่น classification rules จากนั้นจะออกแบบเกณฑ์ในการวัดความน่าสนใจ (interestingness) และความเกี่ยวข้อง (relevancy) ของความรู้ที่ได้จากการทำเหมืองข้อมูล ซึ่งเป็นมาตรฐานหลักในการพิจารณาว่าความรู้ใดบ้างที่มีความถูกต้องและสามารถนำไปใช้ประโยชน์ได้ งานวิจัยนี้จะมีการออกแบบสถาปัตยกรรมของระบบการประมวลผลหลังการทำเหมืองข้อมูล ออกแบบรูปแบบการติดต่อและการส่งความรู้มายังฐานความรู้กลาง ออกแบบวิธีการสร้างและสอบถามความรู้จากฐานความรู้กลาง นอกจากนี้โมเดลในลักษณะของกฎการตัดสินใจ หรือ decision rules จะต้องสามารถนำไปใช้เป็นความรู้ในระบบฐานความรู้ (knowledge-base system) หรือระบบผู้เชี่ยวชาญ (expert system) ได้

ขอบเขตของการวิจัย

โครงการนี้มีจุดมุ่งหมายที่จะพัฒนาแนวทางและเทคนิคในการจัดการและประมวลผลความรู้ที่ได้จากกระบวนการทำเหมืองข้อมูล โดยในเบื้องต้นจะเน้นเฉพาะการทำเหมืองข้อมูลประเภท classification ที่มีการจำแนกข้อมูลเป็นสองคลาส (binary classification) โดยกำหนดข้อมูลเป็นชนิดข้อความ (nominal, categorical)

การจัดการความรู้จะประกอบด้วยการกำหนดเกณฑ์คัดเลือกความรู้ (knowledge evaluation) เพื่อแยกเก็บไว้เฉพาะความรู้ที่เป็นประโยชน์ กำหนดวิธีการรวบรวมความรู้ไว้ในฐานความรู้ (knowledge integration) และการสอบถามความรู้ (knowledge access) โดยจะไม่รวมการ update และการ maintain ฐานความรู้

การพัฒนาโปรแกรมใช้ภาษา Prolog ซึ่งเป็นภาษาเชิงตรรกะ เนื่องจากมีแนวคิดและรูปแบบที่เหมาะสมสำหรับงานที่จะพัฒนาเป็นฐานความรู้ต่อไปในอนาคต ภาษา Prolog ที่ใช้ในงานวิจัยนี้ใช้มาตรฐานของ SWI Prolog (www.swi-prolog.org) ซึ่งเป็นซอฟต์แวร์ประเภทโอเพนซอร์ส (open-source software) ทำให้ผู้ใช้สามารถนำไปใช้งานหรือนำไปพัฒนาต่อได้โดยไม่มีปัญหาเรื่องลิขสิทธิ์ซอฟต์แวร์

ประโยชน์ที่ได้รับจากการวิจัย

สามารถพัฒนากระบวนการวิเคราะห์และตรวจสอบความรู้ รวมถึงการรวบรวมความรู้ที่เป็นผลต่อเนื่องจากการทำเหมืองข้อมูล เพื่อให้การวิจัยในสาขาการทำเหมืองข้อมูลพัฒนาไปสู่รูปแบบการใช้งานที่เป็นอัตโนมัติมากขึ้น และสามารถนำผลของการทำเหมืองข้อมูลมาใช้ประโยชน์โดยสามารถเชื่อมต่อความรู้ในรูปแบบของกฎ เข้ากับฐานความรู้ในระบบผู้เชี่ยวชาญและระบบสนับสนุนการตัดสินใจได้

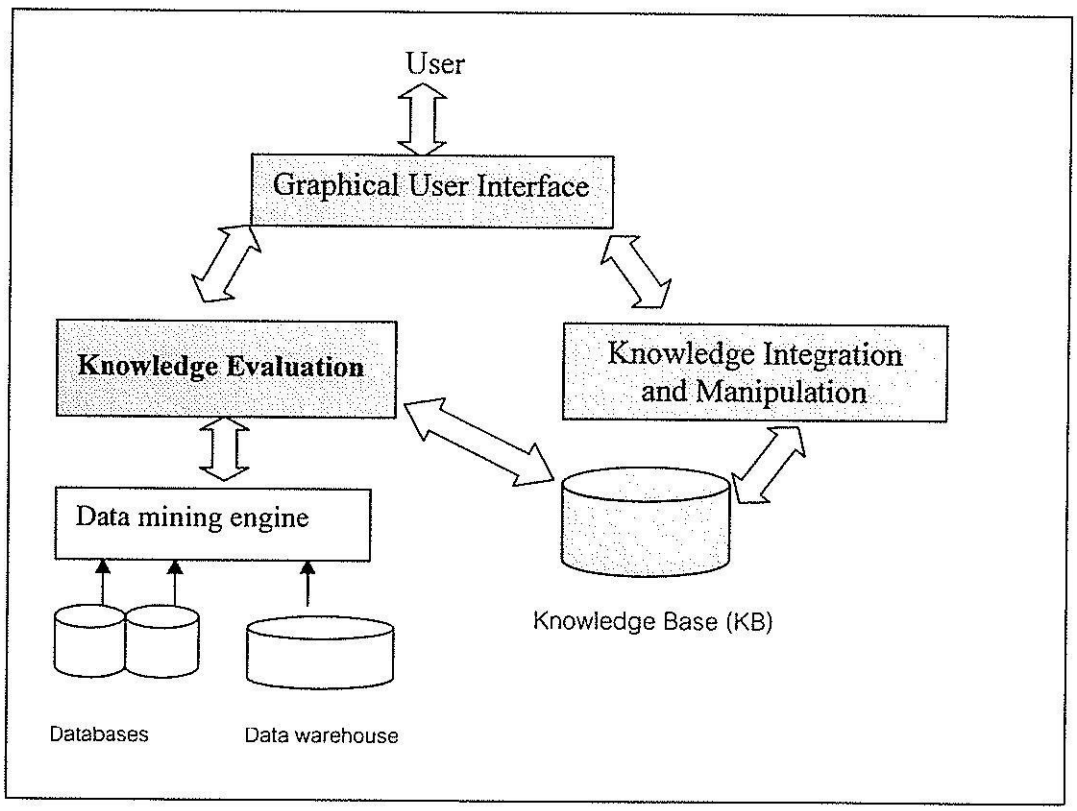
ซอฟต์แวร์การสร้างฐานความรู้อัตโนมัติจากข้อมูลนี้ ผู้วิจัยจะเผยแพร่เป็นสาธารณะผ่านเว็บไซต์ของหน่วยวิจัยด้านวิศวกรรมข้อมูลและการค้นหาความรู้ (Data Engineering and Knowledge Discovery, DEKD) มหาวิทยาลัยเทคโนโลยีสุรนารี ผู้สนใจในทุกหน่วยงานสามารถนำไปใช้ในงานวิเคราะห์ข้อมูลและสร้างฐานความรู้จากผลของการวิเคราะห์ข้อมูลได้

บทที่ 2

วิธีดำเนินการวิจัย

กรอบแนวคิดของงานวิจัย

งานวิจัยนี้มีวัตถุประสงค์หลักเพื่อการพัฒนาเทคนิคในการจัดการกับความรู้ที่เป็นผลลัพธ์จากกระบวนการทำเหมืองข้อมูล ทั้งนี้เนื่องจากผลลัพธ์จากการทำเหมืองข้อมูลมักจะได้ความรู้ที่เป็นโมเดลหรือแพทเทิร์นของข้อมูลในปริมาณมากเกินไป การจัดการกับความรู้ที่ได้จึงหมายถึงการประเมินคุณภาพของความรู้เพื่อคัดเลือกเฉพาะความรู้ที่น่าจะใช้ประโยชน์ได้มากที่สุด โดยระบบต้นแบบที่พัฒนาขึ้นจะประกอบด้วยส่วนประกอบหลักสองส่วนคือ ส่วนประกอบแรกคือส่วนวิเคราะห์ความรู้ (knowledge evaluation) ที่ได้จากการทำเหมืองข้อมูล และส่วนประกอบที่สองคือส่วนรวบรวมและจัดการกับความรู้ (knowledge integration and manipulation) ซึ่งจะรวมถึงการสอบถามความรู้ที่รวบรวมไว้ในฐานความรู้ สถาปัตยกรรมหลักของระบบจัดการความรู้นี้แสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 สถาปัตยกรรมของระบบจัดการความรู้ที่ได้จากการทำเหมืองข้อมูล

ส่วนวิเคราะห์ความรู้ทำหน้าที่รับโมเดลข้อมูลจากการทำเหมืองข้อมูลแบบจำแนก โมเดลข้อมูลนี้จะอยู่ในรูปแบบของต้นไม้ตัดสินใจที่มีโครงสร้างหลักคือโหนดต่างๆ (node) ของต้นไม้และเส้นเชื่อมโหนดทุกเส้น (edge) เมื่อได้รับข้อมูล node และ edge ทั้งหมดจากโปรแกรม data mining engine ส่วนวิเคราะห์ความรู้จะแปลงข้อมูล node และ edge ให้อยู่ในรูปแบบของกฎการตัดสินใจ (decision rule) ซึ่งจะเป็นข้อความในลักษณะ IF...THEN... ในขั้นตอนนี้จะมีการนับจำนวนข้อมูลที่ถูกครอบคลุมโดยกฎแต่ละกฎ แล้วแปลงเป็นค่าสัดส่วนของจำนวนข้อมูลที่กฎครอบคลุมเทียบกับจำนวนข้อมูลทั้งหมด ค่าสัดส่วนที่ได้จะมีค่าอยู่ระหว่าง 0.0-1.0 ดังนั้นค่าสัดส่วนนี้สามารถแปลความหมายได้ว่า เป็นความน่าจะเป็นที่กฎจะประยุกต์ใช้ได้กับเหตุการณ์ที่จะเกิดขึ้นในอนาคต กฎที่มีความน่าจะเป็นสูงจะถูกคัดเลือกและบันทึกไว้ในฐานความรู้

ส่วนรวบรวมและจัดการกับความรู้ ทำหน้าที่จัดเก็บความรู้ในรูปแบบของกฎการตัดสินใจและทำหน้าที่เป็น expert system shell เพื่ออำนวยความสะดวกให้ผู้ใช้สามารถสอบถามความรู้จากฐานความรู้ได้ ผลลัพธ์ที่แสดงให้ผู้ใช้เห็นจะเป็นข้อแนะนำการตัดสินใจพร้อมกับแสดงค่าความน่าจะเป็นของข้อแนะนำนั้น ค่าความน่าจะเป็นนี้จะหมายถึงระดับความเชื่อมั่นของคำแนะนำ

การออกแบบอัลกอริทึมเพื่อการประเมินและคัดเลือกกฎ

ในการออกแบบอัลกอริทึมที่เป็นโมดูลหลักของโปรแกรมการประมวลผลหลังกระบวนการทำเหมืองข้อมูล จะออกแบบสอดคล้องกับสถาปัตยกรรมของระบบที่แสดงดังรูปที่ 2.1 ดังนั้นระบบนี้จะประกอบด้วยสองโมดูล คือ Knowledge evaluation และ Knowledge integration and manipulation รายละเอียดของแต่ละอัลกอริทึมแสดงได้ดังต่อไปนี้

โมดูล Knowledge evaluation

อัลกอริทึมในส่วนนี้ทำหน้าที่รับข้อมูล node และ edge จากโปรแกรมทำเหมืองข้อมูลที่ทำหน้าที่สร้างโมเดลข้อมูลในลักษณะของต้นไม้ตัดสินใจ อัลกอริทึมนี้เริ่มต้นทำงานด้วยการแสดงจอภาพโต้ตอบกับผู้ใช้ให้ระบุชื่อฐานข้อมูลและค่าความน่าจะเป็นขั้นต่ำจากผู้ใช้ ค่านี้จะถูกใช้เป็นเกณฑ์ในการคัดเลือกกฎการตัดสินใจเพื่อบันทึกลงฐานความรู้ โมเดลข้อมูลที่แสดงด้วยโครงสร้าง node และ edge จะถูกแปลงเป็นกฎการตัดสินใจด้วยวิธีการ traverse โครงสร้างต้นไม้ ตั้งแต่โหนดราก (โหนดหมายเลข 0) ไล่ไปตามเส้นเชื่อมต่างๆจนกระทั่งถึงโหนดใบ โครงสร้างข้อมูลของ node และ edge แสดงได้ดังนี้

node(nodeID, [Positive_Instances]-[Negative_Instancess])

edge(ParentNode, EdgeLabel, ChildNode)

โหนดแต่ละโหนดจะประกอบด้วยหมายเลขโหนดและข้อมูลในโหนด การระบุข้อมูลที่อยู่ในโหนดแต่ละโหนดจะใช้วิธีระบุหมายเลขข้อมูล (ข้อมูลแต่ละรายการจะมีหมายเลขกำกับ) โดยข้อมูลจะถูกแยกเป็น positive instances และ negative instances ถ้าโหนดนั้นเป็นโหนดใบและข้อมูลเป็น positive instances ทั้งหมด ลิสต์ของ negative instances จะเป็นลิสต์ว่าง และในทำนองเดียวกันถ้าโหนดนั้นเป็นโหนดใบและข้อมูลในโหนดเป็น negative instances ทั้งหมด ลิสต์ของ positive instances จะเป็นลิสต์ว่าง ดังนั้นการปรากฏลิสต์ใดลิสต์หนึ่งในโหนดเป็นลิสต์ว่างจะหมายถึงการเป็นโหนดใบ

ในส่วนของโครงสร้างกิ่งหรือ edge จะทำหน้าที่เชื่อมโยงโหนดที่เป็นโหนดแม่ไปยังโหนดลูก ดังนั้นโครงสร้าง edge จะระบุข้อมูลสามส่วนคือ หมายเลขโหนดที่เป็นโหนดแม่ของ edge, ชื่อของ edge (คือชื่อและค่าของแอททริบิวต์ เช่น outlook=sunny) และหมายเลขของโหนดที่เป็นโหนดลูก

ผลลัพธ์สุดท้ายที่ได้จากอัลกอริทึมคือกฎการตัดสินใจ หรือ decision rules โดยกฎนี้จะเรียงลำดับตามค่าความน่าจะเป็นสูงสุดลดหลั่งลงมาจนกระทั่งถึงกฎการตัดสินใจที่มีค่าความน่าจะเป็นต่ำสุดตามที่ผู้ใช้กำหนด รายละเอียดขั้นตอนต่างๆในอัลกอริทึมแสดงได้ดังต่อไปนี้

Algorithm 1 Knowledge evaluation

Input: a data model as decision tree with node and edge structures

Output: a set of probabilistic decision rules ranking in descending order

---

(1) Display GUI to get a dataset name and a minimum probability value

(2) Traverse tree from a root node to each leaf node

  (2.1) Collect edge information and count number of data instances

  (2.2) Compute probability as a proportion
  (number of instances at leaf node) / (total data instances in a data set)

  (2.3) Assert a rule containing a triplet
  <attribute-value pair, class, probability value> into temporary KB

(3) Sort rules in the KB in descending order according to the rules' probability

(4) Remove rules that have probability less than the specified threshold

(5) Assert selected rules into the KB and return KB as an output

---

ขั้นตอนแรกของอัลกอริทึมเป็นการอ่านข้อมูลจาก node และ edge เรียงลำดับตามค่าหมายเลขโหนด โดยเริ่มจากโหนดรากที่เป็นโหนดหมายเลขศูนย์ไล่ตามแต่ละกิ่งหรือ edge ลงมา

จนกระทั่งถึงโหนดใบ ในขณะที่อ่านข้อมูลในแต่ละกิ่งจะนับจำนวนข้อมูลที่ถูกแยกย่อยลงมาในแต่ละกิ่งเพื่อคำนวณค่าความน่าจะเป็นในการครอบคลุมข้อมูลของกฎการตัดสินใจ ค่าความน่าจะเป็นนี้เป็นสัดส่วนระหว่างจำนวนข้อมูลที่โหนดใบหารด้วยจำนวนข้อมูลทั้งหมด (นั่นคือหารด้วยข้อมูลที่โหนดราก) จากนั้นบันทึกกฎการตัดสินใจที่แต่ละกฎมีส่วนประกอบ 3 ส่วน คือ ค่าของแอททริบิวต์ที่เป็นปัจจัยประกอบการตัดสินใจ, คลาสของข้อมูลที่เป็นผลของการตัดสินใจ และ ค่าความน่าจะเป็นของกฎการตัดสินใจ

เมื่อแปลงโครงสร้างต้นไม้ตัดสินใจเป็นกฎการตัดสินใจได้ครบในทุกกิ่ง และทุกโหนดใบแล้ว จะมีการเรียงลำดับกฎตามค่าความน่าจะเป็น (ขั้นตอนที่ 3 ตามอัลกอริทึม) โดยเรียงจากค่ามากสุดลงมาหาค่าน้อยที่สุด ต่อจากนั้นจะตัดทิ้งกฎที่มีค่าความน่าจะเป็นต่ำกว่าเกณฑ์ที่ผู้ใช้กำหนด (ขั้นตอนที่ 4 ตามอัลกอริทึม) และในขั้นตอนสุดท้ายเป็นการบันทึกกฎการตัดสินใจที่ได้รับการคัดเลือกแล้วลงในฐานความรู้

โมดูล Knowledge integration and maipulation

อัลกอริทึมในส่วนนี้ทำหน้าที่แปลงกฎการตัดสินใจที่คัดเลือกไว้เฉพาะกฎที่มีค่าความน่าจะเป็นสูงถึงเกณฑ์ที่กำหนด เป็นกฎที่จะใช้ในระบบผู้เชี่ยวชาญ (expert system) เพื่ออำนวยความสะดวกให้ผู้ใช้สามารถสอบถามความรู้ต่างๆ จากฐานความรู้ กฎที่จำเป็นต้องใช้ในระบบผู้เชี่ยวชาญจะประกอบด้วยกฎที่เรียกว่า expert rules และ consulting rules

กฎในกลุ่มของ expert rules จะถูกใช้ในระดับ top-goal ของกระบวนการอนุมาน (inference process) วิธีการแปลงจากกฎการตัดสินใจให้เป็นกฎที่เรียกว่า expert rule จะต้องมีการสร้างส่วน head และส่วน body ของกฎ expert ส่วน head จะเป็นผลการตัดสินใจ (คือ edge label ในโหนดใบของโครงสร้างต้นไม้ หรือส่วน Then ในกฎการตัดสินใจ) ส่วน body จะเป็นรายละเอียดแอททริบิวต์ที่ปรากฏในส่วน If ของกฎการตัดสินใจ

กฎในกลุ่มของ consulting rules จะถูกใช้เพื่อการโต้ตอบและสอบถามรายละเอียดข้อมูล (หมายถึง ค่าของแอททริบิวต์ต่างๆ) จากผู้ใช้ในระหว่างที่ระบบผู้เชี่ยวชาญทำการวิเคราะห์เพื่อหาคำแนะนำที่เหมาะสมแสดงแก่ผู้ใช้ จำนวนกฎที่สร้างจะเท่ากับจำนวนแอททริบิวต์ทั้งหมดของข้อมูล ส่วน head ของกฎจะเป็นชื่อแอททริบิวต์ ส่วน body ของกฎจะเป็นชื่อเพรดิเคตที่มีอาร์กิวเม้นท์เป็นชื่อแอททริบิวต์และค่าที่เป็นไปได้ทั้งหมดของแอททริบิวต์นั้นๆ

รายละเอียดขั้นตอนต่างๆ ของการแปลงกฎการตัดสินใจ ให้เป็น expert rules และ consulting rules เพื่อใช้ในระบบผู้เชี่ยวชาญแสดงได้ดังอัลกอริทึมต่อไปนี้

Algorithm 2 Knowledge integration and manipulation

Input: a set of probabilistic decision rules stored in KB

Output: a set of rules to be used by an expert system shell

(1) For each probabilistic decision rule

  (1.1) Scan information in the If-part and the Then-part

  (1.2) Generate head of expert rule from the Then-part
    type (Then-part, probability value) :-

  (1.3) Generate body of expert rule from the If-part
    :- atrribute_name1(value), ..., attribute_nameN(value).

  (1.4) Write an expert rule in a knowledge-base file, KB_file

(2) For each data attribute

  (2.1) Generate head of consulting rule
    attribute_name(X) :-

  (2.2) Generate body of consulting rule
    :- menuask(attribute_name, X, [list of attribute values]).

(3) Assert consulting rules into the KB_file and return KB as an output

การพัฒนาโปรแกรมเพื่อการประมวลผลหลังการทำเหมืองข้อมูล

โปรแกรมเพื่อการทำเหมืองข้อมูลแบบจำแนกนี้พัฒนาขึ้นโดยใช้ภาษาโปรล็อก การอธิบายขั้นตอนพัฒนาโปรแกรมจะใช้ไวยากรณ์ของภาษาโปรล็อกตามมาตรฐานของ SWI Prolog เวอร์ชัน 5.6.55 (ดาวน์โหลดได้จากเว็บไซต์ www.swi-prolog.org) ลักษณะเด่นประการหนึ่งของภาษาโปรล็อกคือการใช้รูปแบบเดียวกันของทั้งข้อมูลและคำสั่งที่ทำงานกับข้อมูล

รูปแบบของข้อมูล

ข้อมูลที่จะเป็นอินพุตของโปรแกรมทำเหมืองข้อมูล จะมีลักษณะเป็นข้อความที่อยู่ในรูปแบบของข้อความที่เป็นจริง หรือ fact ตัวอย่างของข้อมูลแสดงได้ดังรูปที่ 2.2 ข้อมูลดังรูปเป็นข้อมูลการวินิจฉัยของจักษุแพทย์ว่าคนไข้ที่มารับการวินิจฉัยแต่ละราย สามารถใส่คอนแท็กเลนส์ได้หรือไม่ ถ้าคนไข้สามารถใส่คอนแท็กเลนส์ได้จะมีค่า class=yes ส่วนรายที่ไม่สามารถใส่คอนแท็กเลนส์ (เนื่องจากพยาธิสภาพไม่เหมาะสม เช่นมีนัยน์ตาแห้งเกินไป) จะมีค่า class=no

```
%% Data lens

% attributes: names and their possible values
%
attribute(age, [young, pre_presbyopic, presbyopic]).
attribute(spectacle, [myope, hypermetrope]).
attribute(astigmatism, [no, yes]).
attribute(tear, [reduced, normal]).
attribute(class, [ yes, no]).

% data
instance(1, class=no, [age=young, spectacle=myope, astigmatism=no, tear=reduced]).
instance(2, class=yes, [age=young, spectacle=myope, astigmatism=no, tear=normal]).
instance(3, class=no, [age=young, spectacle=myope, astigmatism=yes, tear=reduced]).
instance(4, class=yes, [age=young, spectacle=myope, astigmatism=no, tear=normal]).
instance(5, class=no, [age=young, spectacle=hypermetrope, astigmatism=no, tear=reduced]).
instance(6, class=yes, [age=young, spectacle=hypermetrope, astigmatism=no, tear=normal]).
instance(7, class=no, [age=young, spectacle=hypermetrope, astigmatism=yes, tear=reduced]).
instance(8, class=yes, [age=young, spectacle=hypermetrope, astigmatism=yes, tear=normal]).
instance(9, class=no, [age=pre_presbyopic, spectacle=myope, astigmatism=no, tear=reduced]).
instance(10,class=yes, [age=pre_presbyopic, spectacle=myope, astigmatism=no, tear=normal]).
instance(11,class=no,[age=pre_presbyopic, spectacle=myope, astigmatism=yes, tear=reduced]).
instance(12,class=yes,[age=pre_presbyopic, spectacle=myope, astigmatism=yes, tear=normal]).
instance(13,class=no,[age=pre_presbyopic,spectacle=hypermetrope,astigmatism=no,tear=reduced]).
instance(14,class=yes,[age=pre_presbyopic,spectacle=hypermetrope,astigmatism=no,tear=normal]).
instance(15,class=no,[age=pre_presbyopic,spectacle=hypermetrope,astigmatism=yes,tear=reduced]).
instance(16,class=no,[age=pre_presbyopic,spectacle=hypermetrope,astigmatism=yes, tear=normal]).
instance(17,class=no, [age=presbyopic, spectacle=myope, astigmatism=no, tear=reduced]).
instance(18,class=no, [age=presbyopic, spectacle=myope, astigmatism=no, tear=normal]).
instance(19,class=no, [age=presbyopic, spectacle=myope, astigmatism=yes, tear=reduced]).
instance(20,class=yes, [age=presbyopic, spectacle=myope, astigmatism=yes, tear=normal]).
instance(21,class=no,[age=presbyopic, spectacle=hypermetrope, astigmatism=no, tear=reduced]).
instance(22,class=yes,[age=presbyopic, spectacle=hypermetrope, astigmatism=no, tear=normal]).
instance(23,class=no,[age=presbyopic, spectacle=hypermetrope, astigmatism=yes, tear=reduced]).
instance(24,class=no,[age=presbyopic, spectacle=hypermetrope, astigmatism=yes, tear=normal]).
%
```

รูปที่ 2.2 ข้อมูลที่ใช้เพื่อสร้างกฎการตัดสินใจ

บรรทัดแรกของข้อมูลเริ่มต้นด้วยเครื่องหมาย % หมายถึง comment โครงสร้างของไฟล์ข้อมูลจะแยกส่วนประกอบออกเป็นสองส่วนคือ ส่วนคำอธิบายแอททริบิวต์ (ได้แก่ส่วนข้อความ attribute ...) และส่วนแสดงรายละเอียดของข้อมูลแต่ละเรคคอร์ด (ได้แก่ส่วนข้อความ instance ...) ในส่วนที่อธิบายแอททริบิวต์ ภายในจะประกอบด้วยสองอาร์กิวเม้นต์ อาร์กิวเม้นต์แรกจะบอกชื่อแอททริบิวต์ อาร์กิวเม้นต์ที่สองเป็นลิสต์ของค่าที่เป็นไปได้ทั้งหมดของแอททริบิวต์นั้น ในส่วนของข้อมูลหรือ instance จะประกอบด้วยสามอาร์กิวเม้นต์ คือ หมายเลขของข้อมูล, ค่าคลาสของข้อมูล, ลิสต์ที่ระบุค่าของแต่ละแอททริบิวต์ โดยวิธีการระบุค่าจะใช้รูปแบบ attribute-value pair หรือ ชื่อแอททริบิวต์=ค่าของแอททริบิวต์ เมื่อสร้างข้อมูลในรูปแบบที่กำหนดเสร็จแล้วจะต้องบันทึกไฟล์ให้อยู่ในรูปแบบของโปรแกรม Prolog ที่มีส่วนขยาย (file extension) เป็น .pl เช่น

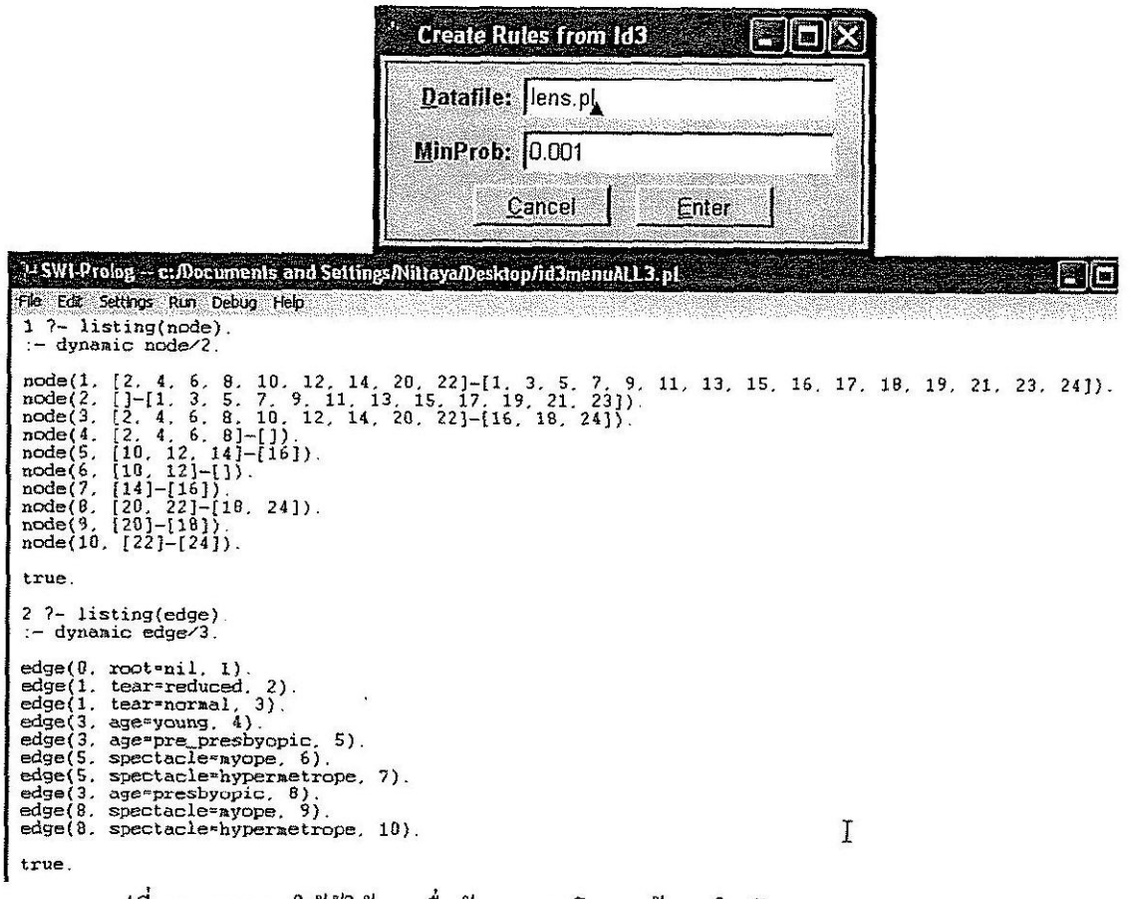
ตัวอย่างไฟล์ข้อมูลในรูปที่ 2.2 บันทึกอยู่ในชื่อ lens.pl ข้อมูลนี้จะใช้เป็นตัวอย่างประกอบคำอธิบายการทำงานของโปรแกรมวิเคราะห์ความรู้ และโปรแกรมรวบรวมและจัดการกับความรู้

โปรแกรมวิเคราะห์ความรู้

การทำงานของโปรแกรมในส่วนนี้ จะเริ่มหลังจากมีการสร้างโมเดลข้อมูลด้วยหลักการของการสร้างต้นไม้ตัดสินใจ (อัลกอริทึม ID3) และบันทึกโมเดลไว้ในเพรดิเคตชื่อ node และ edge คำสั่งต่างๆของโปรแกรมหลักแสดงได้ดังนี้

```
mainId3(Min) :-  init(AllAttr, EdgeList),  % initialize node and edge information
                 getnode(N),               % get node ID
                 create_edge_onelevel(N, AllAttr, EdgeList),  % create tree
                 addAllKnowledge,          % generate decision rules
                 selectRule(Min, Res),     % select top rules
                 writeln(Res),
                 tell('1.knb'),            % write selected decision rules to file
                 writeHeadF,               % transform to expert rules (head)
                 maplist(createRule1, Res),
                 nl, writeTailF,           % generate body of expert rules
                 told,                     % write expert rules to file and close it
                 writeln(endProcess).
```

โปรแกรมหลักชื่อ mainId3 จะรับอาร์กิวเม้นต์ Min ที่ระบุค่าความน่าจะเป็นขั้นต่ำที่ผู้ใช้ต้องการ จากนั้นเริ่มสร้างต้นไม้ตัดสินใจด้วยการเรียกใช้คำสั่งย่อยสามคำสั่งคือ init, getnode และ create_edge_onelevel คำสั่ง create_edge_onelevel จะมีการทำงานซ้ำแบบ recursive เพื่อสร้างต้นไม้ตัดสินใจคราวละหนึ่งระดับ การสร้างต้นไม้จะยุติเมื่อสามารถแยกข้อมูลที่มีสองคลาสปนกันให้เป็นข้อมูลคลาสเดียวกันได้ทั้งหมด หรือเมื่อไม่มีแอททริบิวต์ให้ใช้แยกข้อมูลได้อีกต่อไป จากข้อมูล lens.pl ในรูปที่ 2.2 เมื่อสร้างต้นไม้ตัดสินใจเสร็จ จะได้โครงสร้าง node และ edge ดังรูปที่ 2.3



รูปที่ 2.3 จอภาพให้ผู้ใช้ระบุชื่อข้อมูลและโมเดลข้อมูลในลักษณะ node และ edge

ตัวอย่างในรูปที่ 2.3 ระบุค่าความน่าจะเป็นขั้นต่ำ 0.001 จะทำให้ได้กฎการตัดสินใจจำนวนสามกฎ ดังต่อไปนี้

0.5 >> [tear=reduced] >> no,

0.166667 >> [tear=normal, age=young] >> yes,

0.0833333 >> [tear=normal, age=pre_presbyopic, spectacle=myope] >> yes

โครงสร้างของกฎการตัดสินใจจะประกอบด้วยรายละเอียดสามส่วน แต่ละส่วนแยกจากกันด้วยสัญลักษณ์ >> รายละเอียดส่วนแรกระบุความน่าจะเป็นที่กฎนี้สามารถครอบคลุมข้อมูล รายละเอียดส่วนที่สองระบุลักษณะหรือแอททริบิวต์ที่ใช้ประกอบการตัดสินใจ และส่วนสุดท้ายเป็นผลการวินิจฉัยของแพทย์ ค่า yes หมายถึงแนะนำให้คนไข้ที่มีปัญหาด้านสายตาใส่คอนแท็กเลนส์ได้ ค่า no หมายถึงไม่แนะนำให้ใส่คอนแท็กเลนส์ ตัวอย่างเช่นกฎข้อแรกระบุว่า ถ้าคนไข้มีอัตราการสร้างน้ำตาต่ำ (tear=reduced) แพทย์จะไม่แนะนำให้ใส่คอนแท็กเลนส์ ค่าความน่าจะเป็น 0.5 หมายถึงกฎนี้ครอบคลุมข้อมูล 0.5 หรือ 50% (นั่นคือกฎนี้สังเคราะห์ได้จากข้อมูลคนไข้ 12 ราย จากข้อมูลคนไข้ทั้งหมดจำนวน 24 ราย)

โปรแกรมรวบรวมและจัดการกับความรู้

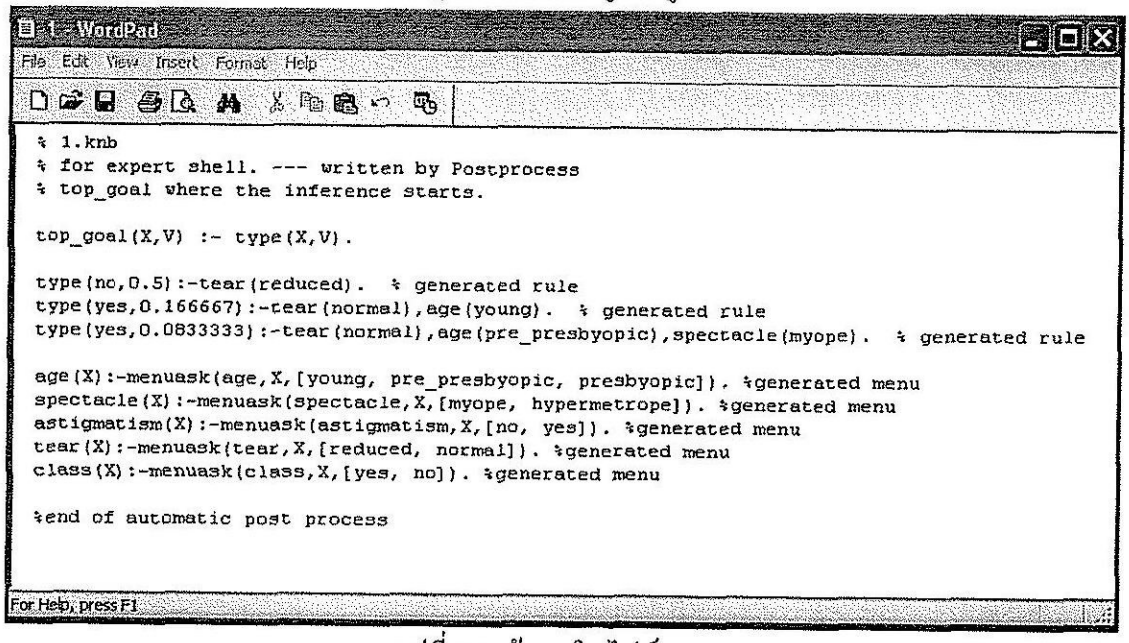
กฎการตัดสินใจที่ได้จากโปรแกรมวิเคราะห์ความรู้ จะถูกนำมาแปลงให้เป็นกฎเพื่อใช้ในระบบผู้เชี่ยวชาญ โปรแกรมแปลงกฎการตัดสินใจให้เป็นกฎประเภท expert rules แสดงได้ดังนี้

```
writeHeadF :-
        format('% 1.knb ~n% for expert shell. --- written by Postprocess'),
        format('~n% top_goal where the inference starts.~n'),
        format('~ntop_goal(X,V) :- type(X,V).~n').

writeTailF:-
        findall(_,(attribute(S,L),
                format('~n~w(X):-menuask(~w,X,~w). %generated menu',[S,S,L]))
              ,_),
        format('~n~n%end of automatic post process').
```

โปรแกรมแปลงกฎการตัดสินใจให้เป็นกฎประเภท consult rules แสดงได้ดังนี้

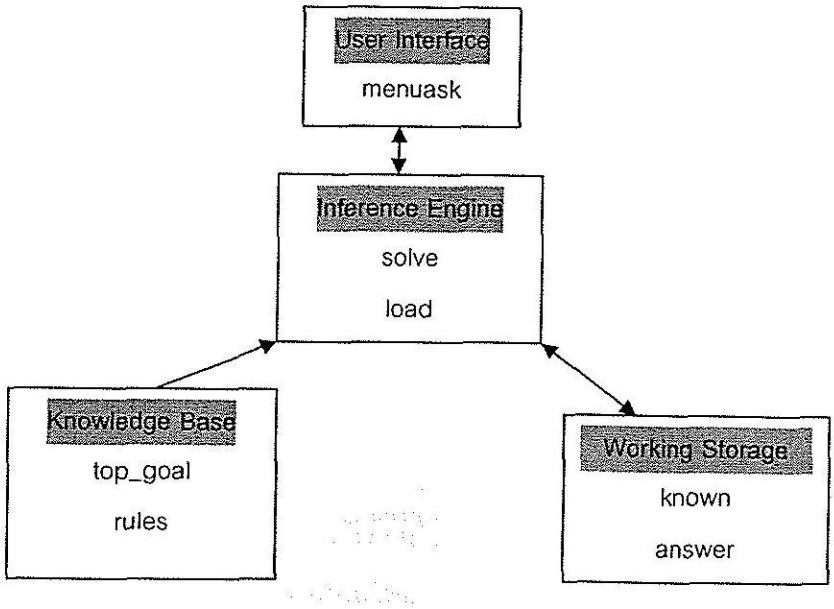
```
transform1([X=V], [Res]) :-
        atomic_list_concat([X,'(',V,')'], Res1),
        term_to_atom(Res, Res1),!.

transform1([X=V|T], [Res|T1]) :-
        atomic_list_concat([X,'(',V,')'], Res1),
        term_to_atom(Res, Res1),
        transform1(T, T1).

createRule1(I) :- I=Z>>X>>Y,
        transform1(X,BodyL),
        format('~ntype(~w,~w):-',[Y,Z]),
        myformat(BodyL) , write('  % generated rule'),!.

myformat([X]) :- write(X), write('.'),!.
myformat([H|T]) :- write(H), write(','), myformat(T).
```

เมื่อโปรแกรมที่ทำหน้าที่แปลงจากกฎการตัดสินใจ ให้เป็นกฎที่สามารถใช้งานได้กับระบบผู้เชี่ยวชาญ ทำการแปลงกฎทั้งหมดเสร็จสิ้น จะบันทึกผลลัพธ์ที่ได้ลงในไฟล์ชื่อ '1.knb' ซึ่งจากตัวอย่างข้อมูล lens.pl เมื่อแปลงกฎเสร็จจะได้ข้อมูลดังรูปที่ 2.4



รูปที่ 2.4 ข้อมูลในไฟล์ 1.knb

โปรแกรม expert system shell จะทำหน้าที่อนุมานจาก expert rules และ consult rules เพื่อแสดงเป็นคำแนะนำแก่ผู้ใช้ โครงสร้างของ expert system shell รวมถึงคำสั่งที่ใช้ในแต่ละส่วนแสดงดังรูปที่ 2.5



รูปที่ 2.5 โครงสร้างของ expert system shell

โครงสร้างของ expert system shell ประกอบด้วยส่วน User interface ทำหน้าที่โต้ตอบในลักษณะ interactive กับผู้ใช้ คำสั่งที่เขียนขึ้นเพื่อใช้กับการทำงานในส่วนนี้คือคำสั่ง menuask ส่วน Inference engine ทำหน้าที่รับข้อมูลบางส่วนจากผู้ใช้เพื่ออนุมานจาก expert rules ในฐานความรู้และหาคำแนะนำที่เหมาะสมที่สุดแสดงแก่ผู้ใช้ ส่วน Inference engine จะติดต่อกับ Working storage ในระหว่างที่ค้นหากฎที่เกี่ยวข้องกับสิ่งที่ผู้ใช้ถาม โดยในส่วน Working storage จะมีคำสั่ง known และ answer ในส่วนของ Inference engine จะประกอบด้วยคำสั่ง load ทำหน้าที่เรียกใช้ไฟล์ที่เกี่ยวข้องในฐานความรู้ และคำสั่ง solve ที่ผู้ใช้สั่งเพื่อให้ระบบผู้เชี่ยวชาญแสดงคำแนะนำ ในส่วนของฐานความรู้จะประกอบด้วยคำสั่งระดับบนสุดเรียกว่า top_goal และกฎต่างๆ เรียกว่า rules ในกรณีที่ผู้ใช้ต้องการทราบเหตุผลหรือคำอธิบายระบบผู้เชี่ยวชาญจะมีคำสั่ง why ให้เรียกใช้เพื่อแสดงคำอธิบายประกอบการให้คำแนะนำ

คำสั่งต่างๆใน expert system shell แสดงได้ดังต่อไปนี้

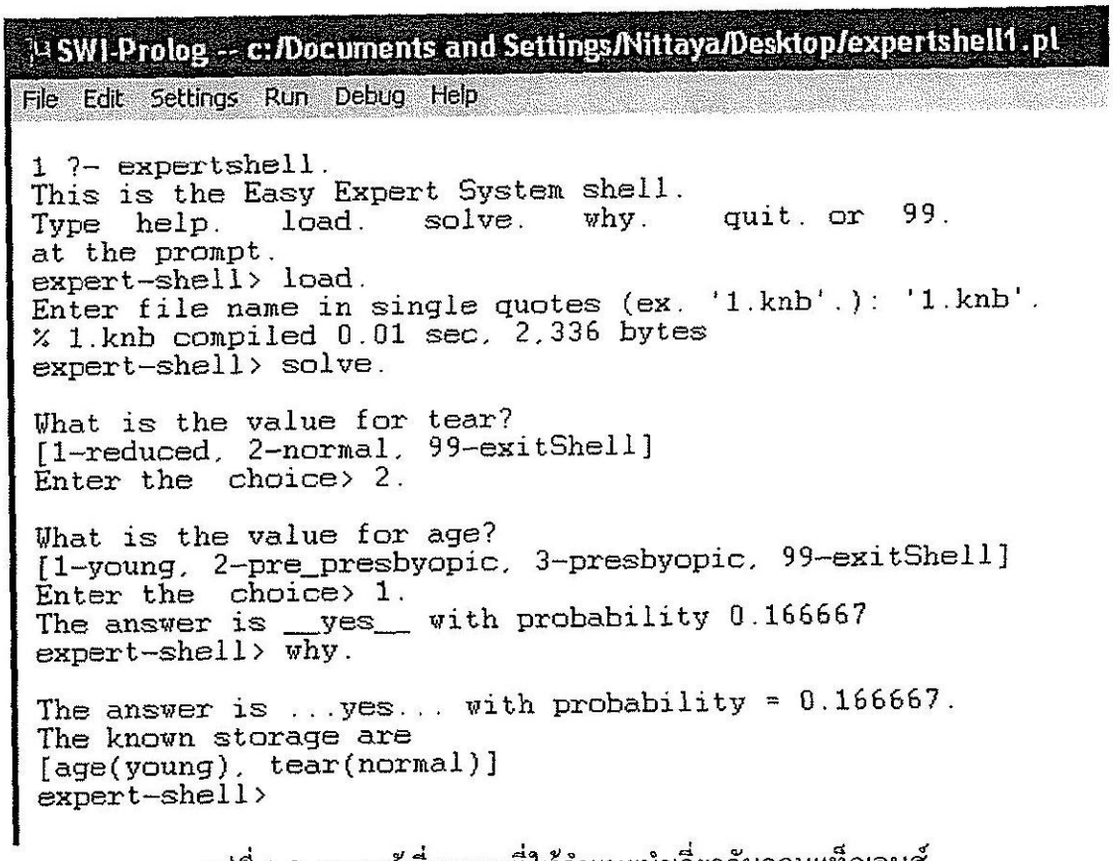
```
expertshell :-   greeting,
                 repeat,
                 write('expert-shell> '),
                 read(X),
                 do(X),
                 (X == quit ; X == 99),
                 writeln('>>>>Goodbye, see you later<<<<'), !.

greeting :-  write('This is the Easy Expert System shell.'), nl,
             native_help.
do(help) :- native_help, !.
do(load) :- load_kb, !.
do(solve) :- solve, !.
do(why) :- why,!.
do(quit).
do(99).
do(X) :-  write(X),
          write(' is not a legal command.'), nl,
          fail.
```

```
native_help :-   write('Type help. load. solve. why. quit. or 99.'),nl,
                 write('at the prompt.'), nl.
load_kb :- write('Enter file name in single quotes (ex. "1.knb".): '),
           read(F),
           reconsult(F).

solve :-  retractall(known( _) ),
          retractall(answer(_,_)),
          top_goal(X,V),
          format('The answer is __~w__ with probability ~w',[X,V]),
          assert(answer(X,V)), nl.
solve :-  write('No answer found.'),nl.

menuask(Pred, Value, Menu) :-
          menuask(Pred,Menu),
          atomic_list_concat([Pred,'(',Value,')'],X),
          term_to_atom(T,X),known(T),!.
menuask(Pred,_) :-
          atomic_list_concat([Pred,'(','_',')'],X),  % check for recorded predicate
          term_to_atom(T,X),known(T),!.  % not ask again

menuask(Attribute,Menu):-
          nl, write('What is the value for '), write(Attribute), write('?'), nl,
          addchoice(Menu,MenuRes),
          writeln(MenuRes), write('Enter the  choice> '), read(C),
          member(C-V,MenuRes),
          (C=99 -> abort ; true ),
          atomic_list_concat([Attribute,'(',V,')'],X),
          term_to_atom(T,X),
          asserta(known(T)) .
```

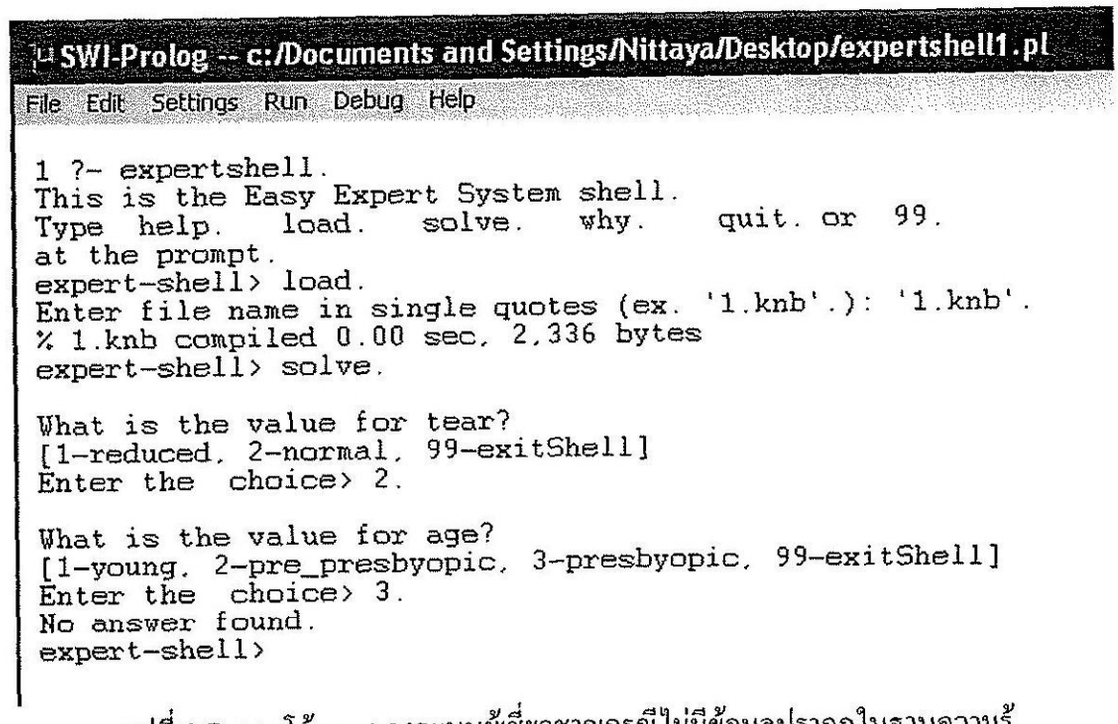
```
why :-    answer(A,V),
          format('~nThe answer is ...~w... with probability = ~w.~n',[A,V]),
          findall( X , known(X),Result),
          writeln('The known storage are'),
          writeln(Result).

addchoice(X,Res):-  length(X,Len),
                    numlist(1,Len,NumL),
                    map(NumL,X,Res).
map([], [], [99-exitShell]).
map([H|T], [X|TT], [H-X|T1]) :- map(T, TT, T1).
```

ตัวอย่างการใช้งานระบบผู้เชี่ยวชาญเพื่อขอคำแนะนำเกี่ยวกับใช้คอนแท็กเลนส์ แสดงได้ดังรูปที่ 2.6 เมื่อรันโปรแกรม expertshell1.pl ผู้ใช้จะเริ่มต้นใช้งาน expert system shell ด้วยการพิมพ์คำสั่ง expertshell (ทุกคำสั่งในภาษาโปรล็อกจะต้องจบคำสั่งด้วยเครื่องหมาย '.') เมื่อ expert system shell เริ่มทำงาน ที่ต้นบรรทัดจะปรากฏข้อความ expert-shell> เพื่อเตรียมรับคำสั่งจากผู้ใช้ คำสั่งแรกของการใช้งานคือคำสั่ง load เพื่อเรียกใช้ไฟล์ฐานความรู้ที่เกี่ยวข้อง ซึ่งในตัวอย่างนี้ใช้ไฟล์ '1.knb' จากนั้นใช้คำสั่ง solve โปรแกรมจะเริ่มถามข้อมูลต่างๆจากผู้ใช้ เมื่อได้ข้อมูลที่ต้องการเพียงพอแล้ว จะแสดงคำแนะนำให้ผู้ใช้ทราบ พร้อมทั้งค่าความน่าจะเป็นเพื่อให้ผู้ใช้ทราบว่าจะเชื่อมั่นในคำแนะนำนั้นได้มากน้อยเพียงใด และถ้าผู้ใช้ต้องการคำอธิบายประกอบคำแนะนำสามารถเรียกดูคำอธิบายได้โดยการพิมพ์คำสั่ง why

ในกรณีที่ข้อมูลที่ผู้ใช้ระบุไม่ปรากฏในกฎใดๆ ของฐานความรู้ ระบบผู้เชี่ยวชาญจะโต้ตอบด้วยข้อความดังรูปที่ 2.7



รูปที่ 2.6 ระบบผู้เชี่ยวชาญที่ให้คำแนะนำเกี่ยวกับคอนแท็กเลนส์



รูปที่ 2.7 การโต้ตอบของระบบผู้เชี่ยวชาญกรณีไม่มีข้อมูลปรากฏในฐานความรู้

บทที่ 3
การทดสอบโปรแกรม

วิธีการทดสอบโปรแกรมเพื่อการประมวลผลหลังการทำเหมืองข้อมูล

การทำเหมืองข้อมูลในงานวิจัยนี้เน้นที่การทำเหมืองข้อมูลเพื่อการจำแนก วิธีการสร้างโมเดลข้อมูลใช้หลักการของการสร้างต้นไม้ตัดสินใจตามอัลกอริทึม ID3 (Quinlan 1993) ที่ใช้การสร้างโหนดของต้นไม้เป็นเครื่องมือในการแยกข้อมูลที่มีหลายคลาสปะปนกัน ให้เหลือเป็นกลุ่มข้อมูลย่อยที่เป็นคลาสเดียวกัน จากนั้นอ่านลักษณะของข้อมูลในแต่ละกลุ่มย่อยจากโครงสร้างโหนดและกิ่งของต้นไม้ตัดสินใจ วิธีการจำแนกข้อมูลเพื่อให้ได้โมเดลในลักษณะนี้ได้รับการยอมรับว่ามีความถูกต้องของโมเดลสูง และข้อเด่นของวิธีการนี้คือโมเดลเข้าใจได้ง่าย

โมเดลที่ได้จากการทำเหมืองข้อมูลนี้จะถูกประมวลผลต่อไป โดยโมเดลจะถูกเปลี่ยนเป็นกฎการตัดสินใจที่มีค่าความน่าจะเป็นกำกับ ค่าความน่าจะเป็นนี้ถูกใช้เป็นเกณฑ์เพื่อวัดอัตราส่วนของจำนวนข้อมูลที่สามารถอธิบายได้ด้วยกฎการตัดสินใจ (หรือเรียกว่าค่า coverage) ซึ่งค่านี้จะแปลความหมายได้ถึงโอกาสที่กฎนั้นจะประยุกต์ใช้ได้กับข้อมูลใหม่ที่จะเกิดขึ้นในอนาคต

ดังนั้นวิธีการทดสอบความสามารถของโปรแกรมเพื่อการประมวลผลหลังการทำเหมืองข้อมูลจะใช้วิธีแยกข้อมูลส่วนใหญ่เป็นข้อมูลฝึก เพื่อสร้างโมเดลข้อมูลและสร้างกฎการตัดสินใจที่จะถูกแปลงเป็นกฎในระบบผู้เชี่ยวชาญ จากนั้นจะทดสอบความถูกต้องในการให้คำแนะนำของระบบผู้เชี่ยวชาญ โดยข้อมูลที่ใช้ทดสอบเป็นข้อมูลขนาดเล็กมีจำนวน 16 รายการ ชุดข้อมูลที่ใช้ในการทดสอบโปรแกรมนี้เป็นข้อมูลมาตรฐานของ UCI Machine Learning Repository: Data Sets (http://archive.ics.uci.edu/ml/datasets.html) โดยเลือกใช้ข้อมูล post-operative patients (จำนวนข้อมูลฝึกมี 70 เรคคอร์ด แต่ละเรคคอร์ดประกอบด้วย 8 แอททริบิวต์) และข้อมูล breast-cancer recurrences (จำนวนข้อมูลฝึกมี 175 เรคคอร์ด แต่ละเรคคอร์ดประกอบด้วย 9 แอททริบิวต์) ข้อมูลฝึกและข้อมูลทดสอบของชุดข้อมูลทั้งสองในรูปแบบของโปรล็อก ปรากฏในภาคผนวก ข

ตัวอย่างวิธีการทดสอบความถูกต้องของระบบผู้เชี่ยวชาญกับข้อมูลคนไข้รายที่ 71 ในชุดข้อมูล post-operative patients แสดงได้ดังรูปที่ 3.1 ขั้นตอนการใช้งาน expert system shell ที่แสดงในรูปเป็นการสอบถามคำแนะนำกรณีคนไข้หลังผ่าตัด มีค่าอุณหภูมิภายในร่างกาย ค่าความดัน และค่าอื่นๆ เป็น [internalTemp= mid, surfaceTemp=mid, oxygenSaturation=excellent, bloodPressure=high, tempStability=stable, coreTempStability=stable, bpStability=stable, comfort=10] ผลการวินิจฉัยของแพทย์ระบุให้ส่งคนไข้รายนี้ไปที่หอผู้ป่วยหรือ ward และผลจากคำแนะนำของระบบผู้เชี่ยวชาญเป็น ward (คำแนะนำอนุมานจากกฎที่ 12 ในฐานความรู้ของระบบ

ผู้เชี่ยวชาญ) แสดงว่าคำแนะนำที่ระบบผู้เชี่ยวชาญให้แก่คนไข้รายนี้ถูกต้อง กฎที่ใช้ในระบบผู้เชี่ยวชาญตามตัวอย่างนี้ สร้างจากโปรแกรมคัดเลือกกฎการตัดสินใจที่กำหนดค่าความน่าจะเป็นขั้นต่ำ 0.001 กฎทั้งหมดในระบบผู้เชี่ยวชาญแสดงได้ดังรูปที่ 3.2 ในกรณีของข้อมูล breast-cancer recurrences ฐานความรู้ของระบบผู้เชี่ยวชาญแสดงดังรูปที่ 3.3

```
/* *** Test Data
%     =========
instance(71, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
```

```
% 1.knb
% for expert shell. --- written by Postprocess
% top_goal where the inference starts.

top_goal(X,V) :- type(X,V).

type(ward,0.1):-comfort(10),bloodPressure(high),surfaceTemp(low).  % generated rule
type(ward,0.0714286):-comfort(10),bloodPressure(mid),surfaceTemp(low),bpStability(stable).  % generated rule
type(ward,0.0714286):-comfort(10),bloodPressure(mid),surfaceTemp(high),bpStability(mod_stable).  % generated rul
type(ward,0.0571429):-comfort(10),bloodPressure(high),surfaceTemp(mid),bpStability(mod_stable).  % generated rul
type(ward,0.0428571):-comfort(15),bpStability(unstable),surfaceTemp(mid).  % generated rule
type(ward,0.0428571):-comfort(10),bloodPressure(mid),surfaceTemp(mid),bpStability(stable),internalTemp(mid),temp
type(ward,0.0428571):-comfort(10),bloodPressure(low).  % generated rule
type(ward,0.0285714):-comfort(15),bpStability(unstable),surfaceTemp(high).  % generated rule
type(ward,0.0285714):-comfort(15),bpStability(stable),internalTemp(mid),surfaceTemp(mid).  % generated rule
type(ward,0.0285714):-comfort(15),bpStability(mod_stable).  % generated rule
type(home,0.0285714):-comfort(10),bloodPressure(mid),surfaceTemp(mid),bpStability(mod_stable).  % generated rule
type(ward,0.0285714):-comfort(10),bloodPressure(high),surfaceTemp(mid),bpStability(stable),oxygenSaturation(exce
type(home,0.0142857):-comfort(15),bpStability(unstable),surfaceTemp(low).  % generated rule
type(home,0.0142857):-comfort(15),bpStability(stable),internalTemp(mid),surfaceTemp(low),oxygenSaturation(good)
```

```
SWI-Prolog -- c:/Documents and Settings/Nittaya/Desktop/expertshell1.pl
File  Edit  Settings  Run  Debug  Help
expert-shell> solve.

What is the value for comfort?
[1-5, 2-7, 3-10, 4-15, 99-exitShell]
Enter the  choice> 3.

What is the value for bloodPressure?
[1-high, 2-mid, 3-low, 99-exitShell]
Enter the  choice> 1.

What is the value for surfaceTemp?
[1-mid, 2-high, 3-low, 99-exitShell]
Enter the  choice> 1.

What is the value for bpStability?
[1-stable, 2-mod_stable, 3-unstable, 99-exitShell]
Enter the  choice> 1.

What is the value for oxygenSaturation?
[1-excellent, 2-good, 3-fair, 4-poor, 99-exitShell]
Enter the  choice> 1.
The answer is __ward__ with probability 0.0285714
expert-shell> why.

The answer is ...ward... with probability = 0.0285714.
The known storage are
[oxygenSaturation(excellent), bpStability(stable), surfaceTemp(mid), bloodPressure(high), comfor
expert-shell>
```

รูปที่ 3.1 ตัวอย่างการทดสอบความถูกต้องของระบบผู้เชี่ยวชาญ

รูปที่ 3.2 กฎทั้งหมดของระบบผู้เชี่ยวชาญที่สร้างด้วยเกณฑ์ความน่าจะเป็นขั้นต่ำ 0.001

รูปที่ 3.3 ฐานความรู้ของระบบผู้เชี่ยวชาญในการแนะนำเกี่ยวกับ breast-cancer recurrences

การทดสอบความถูกต้องในการให้คำแนะนำของระบบผู้เชี่ยวชาญที่พัฒนาขึ้นในงานวิจัยนี้ จะเปรียบเทียบกับผลการทดสอบในการจำแนกข้อมูลด้วยอัลกอริทึม ID3 (decision-tree induction algorithm), PRISM (rule induction algorithm) และ neural network (multi-layer perceptron algorithm)

ผลการทดสอบโปรแกรม

ผลการทดสอบความถูกต้องในการให้คำแนะนำของระบบผู้เชี่ยวชาญในกรณีของข้อมูล post-operative patients แสดงดังตารางที่ 3.1 และผลการทดสอบความถูกต้องในการให้คำแนะนำของระบบผู้เชี่ยวชาญในกรณีของข้อมูล breast-cancer recurrences แสดงดังตารางที่ 3.2

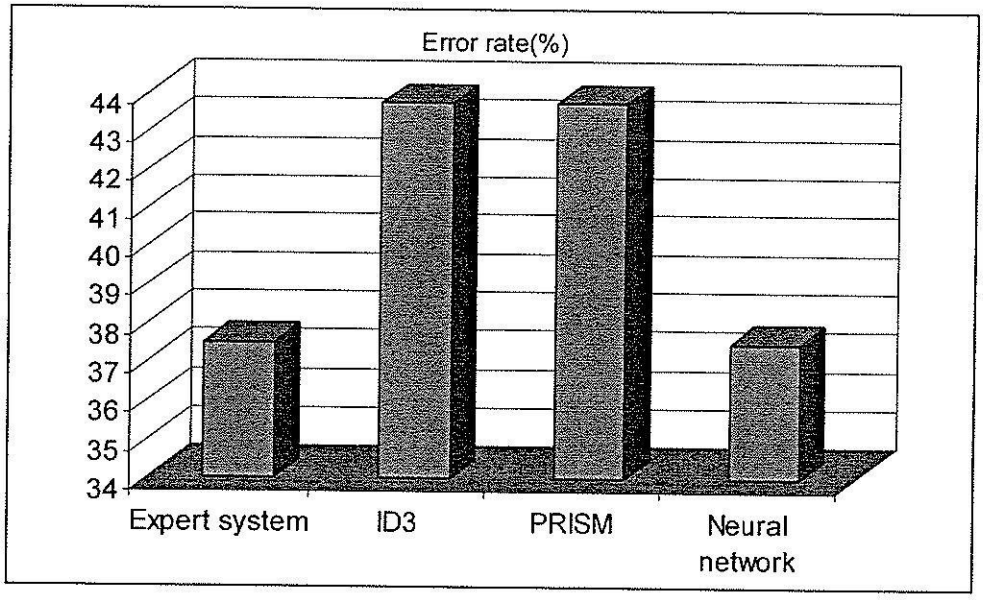
ตารางที่ 3.1 ผลการทดสอบความถูกต้องของระบบผู้เชี่ยวชาญกับข้อมูล post-operative patients

ข้อมูลที่	การวินิจฉัยของแพทย์	คำแนะนำของระบบผู้เชี่ยวชาญ	ผลการทำนายของ ID3	ผลการทำนายของ PRISM	ผลการทำนายของ Neural network
1	Ward	Ward	Ward	Ward	Ward
2	Ward	Ward	Ward	Ward	Ward
3	Ward	Ward	Unclassified	**Home**	Ward
4	Ward	**Home**	**Home**	**Home**	**Home**
5	Ward	Ward	Ward	Unclassified	Ward
6	Home	**Ward**	**Ward**	**Home**	**Home**
7	Ward	**Home**	**Home**	**Home**	**Home**
8	Ward	Ward	Ward	Ward	Ward
9	Home	No answer found	**Ward**	**Ward**	**Ward**
10	Ward	**Home**	**Home**	**Home**	**Home**
11	Ward	Ward	Ward	Unclassified	Ward
12	Ward	Ward	Ward	Ward	Ward
13	Home	**Ward**	**Ward**	**Ward**	**Ward**
14	Ward	Ward	Ward	Ward	Ward
15	Ward	Ward	Ward	Ward	Ward
16	Home	**Ward**	**Ward**	**Ward**	**Ward**

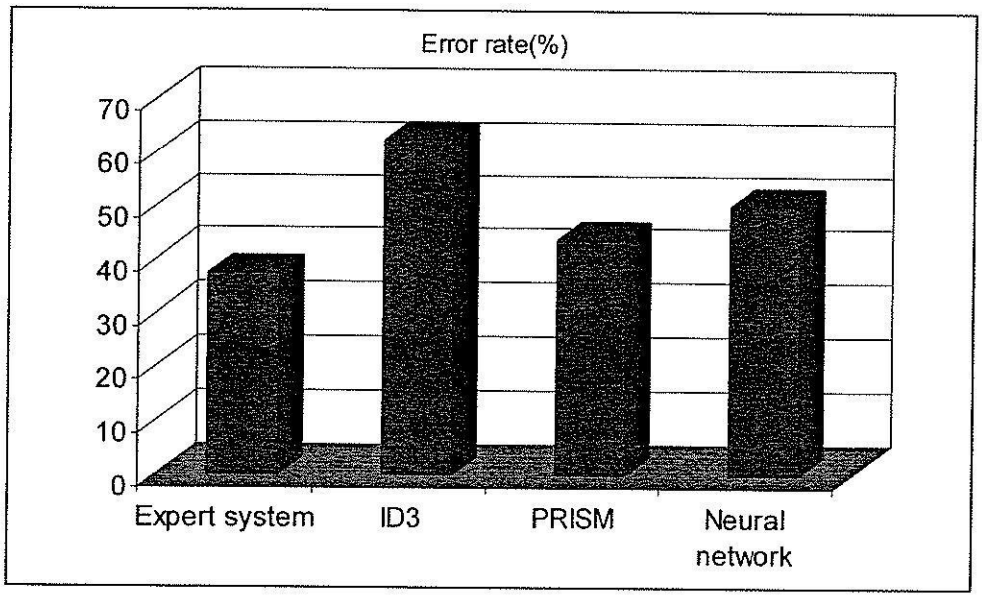
ตารางที่ 3.2 ผลการทดสอบความถูกต้องของระบบผู้เชี่ยวชาญกับข้อมูล breast-cancer recurrences

ข้อมูลที่	การวินิจฉัยของแพทย์	คำแนะนำของระบบผู้เชี่ยวชาญ	ผลการทำนายของ ID3	ผลการทำนายของ PRISM	ผลการทำนายของ Neural network
1	No	No	Yes	No	Yes
2	No	Yes	Yes	Yes	Yes
3	Yes	Yes	No	No	No
4	Yes	No answer found	No	No	Yes
5	No	Yes	Unclassified	No	No
6	No	No answer found	Yes	No	No
7	Yes	No	No	No	No
8	No	No	Yes	Yes	No
9	Yes	No	No	Yes	No
10	No	Yes	No	No	Yes
11	No	No	No	No	No
12	No	Yes	Yes	Yes	Yes
13	Yes	Yes	No	No	Yes
14	No	No answer found	No	No	No
15	No	No	Unclassified	No	Yes
16	Yes	Yes	Yes	Yes	Yes

การทดสอบความอ่อนไหว (sensitivity) ใช้วิธีการพิจารณาความผิดพลาดในการทำนายผลของโมเดล โดยในการเปรียบเทียบมีการทำซ้ำสิบครั้งด้วยวิธีการของ 10-fold cross validation เทคนิคการเปรียบเทียบแบบนี้จะแบ่งข้อมูลออกเป็นสิบส่วน ในแต่ละรอบของการทดสอบ ข้อมูลหนึ่งส่วนจะถูกกันไว้ทำหน้าที่เป็นข้อมูลชุดทดสอบ ในขณะที่ข้อมูลอื่นอีกเก้าส่วนจะทำหน้าที่เป็นข้อมูลฝึกเพื่อสร้างโมเดล เมื่อทำซ้ำครบสิบรอบข้อมูลทุกส่วนจะมีโอกาสทำหน้าที่เป็นข้อมูลทดสอบ ผลการเปรียบเทียบความผิดพลาดในการให้คำแนะนำ หรือการทำนายคลาสของข้อมูลทดสอบของโปรแกรม expert system, ID3, PRISM, Neural network กับข้อมูล post-operative patients และ breast-cancer recurrences ด้วยเทคนิคการเปรียบเทียบแบบ 10-fold cross validation แสดงดังกราฟในรูปที่ 3.4 และ 3.5 ตามลำดับ



รูปที่ 3.4 กราฟเกรียบเทียบ error rate เมื่อทดสอบกับข้อมูล post-operative patients



รูปที่ 3.5 กราฟเกรียบเทียบ error rate เมื่อทดสอบกับข้อมูล breast-cancer recurrences

อภิปรายผล

จากผลการทดสอบความถูกต้องของโปรแกรมประมวลผลหลังการทำเหมืองข้อมูล หรือโปรแกรม expert system ในทั้งสองชุดข้อมูลเมื่อเทียบกับโปรแกรม ID3 พบว่าโปรแกรม expert system ให้ผลลัพธ์ที่ถูกต้องมากกว่าทั้งๆที่โปรแกรม expert system ใช้โมเดลข้อมูลเริ่มต้นเป็นต้นไม้ตัดสินใจเหมือนกับโปรแกรม ID3 แต่ข้อแตกต่างอยู่ที่ โปรแกรม expert system มีการจัดลำดับกฎการตัดสินใจตามค่าความน่าจะเป็นและคัดเลือกใช้เฉพาะกฎที่มีค่าความน่าจะเป็นสูง

เมื่อเปรียบเทียบความผิดพลาดในการให้คำแนะนำ หรือความผิดพลาดในการทำนาย (error rate) ของโปรแกรมทั้งสี่พบว่าโปรแกรม expert system ให้ค่าความผิดพลาดที่ต่ำที่สุด (คือ 37.50% ในข้อมูลทั้งสองชุด) และเมื่อวิเคราะห์ความผิดพลาดในลักษณะของ false negative (ดังตารางที่ 3.3) พบว่าโปรแกรม expert system ให้ความผิดพลาดชนิดนี้ต่ำที่สุด ซึ่งในการวินิจฉัยทางด้านการแพทย์ความผิดพลาดประเภท false negative (เช่นคนไข้เป็นมะเร็ง แต่ได้รับการวินิจฉัยว่าสุขภาพแข็งแรงเป็นปกติ) ถือว่ามีความร้ายแรงมากกว่าความผิดพลาดประเภท false positive (เช่นคนไข้ที่ไม่เป็นมะเร็ง แต่ได้รับการวินิจฉัยว่าเป็นมะเร็งและจะต้องถูกส่งไปตรวจทางคลินิกเพิ่มเติม)

ตารางที่ 3.3 ผลการวิเคราะห์ความผิดพลาดในลักษณะของ false negative

False negative error	Expert system	ID3	PRISM	Neural network
ข้อมูล post-operative patients ลักษณะการทำนายผิด จากส่งกลับหอผู้ป่วย (ward) เป็นส่งกลับบ้าน (home)	Error = 3/16 = 18.75%	Error = 3/16 = 18.75%	Error = 4/16 = 25%	Error = 3/16 = 18.75%
ข้อมูล breast-cancer recurrences ลักษณะการทำนายผิด จากเป็นมะเร็งซ้ำ (yes) เป็นไม่เป็นมะเร็ง (no)	Error = 2/16 = 12.50%	Error = 5/16 = 31.25%	Error = 4/16 = 25%	Error = 3/16 = 18.75%

โปรแกรมที่ใช้โมเดลข้อมูลในลักษณะของต้นไม้ตัดสินใจ (ID3) หรือในลักษณะของกฎ (expert system และ PRISM) จะมีข้อบกพร่องที่เห็นได้ชัดเจนจากผลการทดลองคือ โมเดลที่ได้ไม่สมบูรณ์ เนื่องจากจะมีบางกรณีที่ไม่สามารถให้คำแนะนำ หรือไม่สามารถทำนายคลาสของข้อมูลได้ ซึ่งจะต่างจากโปรแกรม Neural network ที่มีพื้นฐานการสร้างโมเดลจากการใช้ฟังก์ชันทางคณิตศาสตร์ (ในการทดลองนี้ใช้ sigmoid function) ทำให้สามารถทำนายคลาสของข้อมูลได้ครอบคลุมหมดทุกกรณี

บทที่ 4
บทสรุป

สรุปผลการวิจัย

โปรแกรมเพื่อการทำเหมืองข้อมูลในปัจจุบัน มักจะได้รับการพัฒนาโปรแกรมจนถึงขั้นให้สามารถแสดงผลลัพธ์เป็นโมเดลข้อมูล หรือแพทเทิร์นของกลุ่มข้อมูล ขั้นตอนต่อจากนั้นที่เป็นการนำโมเดลหรือแพทเทิร์นไปใช้ประโยชน์มักจะให้อยู่ในดุลพินิจของนักวิเคราะห์ข้อมูล ซึ่งนักวิเคราะห์ข้อมูลที่ไม่คุ้นเคยกับกระบวนการทำเหมืองข้อมูล มักจะประสบกับอุปสรรคสำคัญของการคัดเลือกความรู้ในโมเดลไปใช้งานต่อไป เนื่องจากโปรแกรมทำเหมืองข้อมูลมักจะแสดงผลลัพธ์เป็นโมเดลที่มีความซับซ้อนและมีขนาดใหญ่ทำให้แปลผลได้ยาก ตัวอย่างเช่นจากข้อมูล post-operative patients เมื่อสร้างโมเดลข้อมูลด้วยโปรแกรม ID3 จะได้ผลลัพธ์ที่เป็นโครงสร้างต้นไม้ขนาดใหญ่ (รูปที่ 4.1) และข้อมูลเดียวกันนี้เมื่อสร้างโมเดลด้วยโปรแกรม Multi-layer percentron ซึ่งเป็นโปรแกรมในกลุ่ม Neural network จะได้ผลลัพธ์ (รูปที่ 4.2) เป็นโมเดลข้อมูลในลักษณะฟังก์ชันที่แปลผลได้ค่อนข้างยากสำหรับผู้ที่ไม่คุ้นเคย

```
comfort = 05: home
comfort = 07: home
comfort = 10
| bloodPressure = high
| | surfaceTemp = mid
| | | bpStability = stable
| | | | oxygenSaturation = excellent: ward
| | | | oxygenSaturation = good
| | | | | surfaceTempStability = stable: home
| | | | | surfaceTempStability = mod-stable: null
| | | | | surfaceTempStability = unstable: ward
| | | | oxygenSaturation = fair: null
| | | | oxygenSaturation = poor: null
| | | bpStability = mod-stable: ward
| | | bpStability = unstable: home
| | surfaceTemp = high
| | | internalTemp = mid: home
| | | internalTemp = high: ward
| | | internalTemp = low: null
| | surfaceTemp = low: ward
| bloodPressure = mid
| | surfaceTemp = mid
| | | bpStability = stable
| | | | internalTemp = mid
| | | | | surfaceTempStability = stable: ward
| | | | | surfaceTempStability = mod-stable: null
| | | | | surfaceTempStability = unstable: ward
| | | | internalTemp = high: home
| | | | internalTemp = low
| | | | | oxygenSaturation = excellent: home
| | | | | oxygenSaturation = good: home
| | | | | oxygenSaturation = fair: null
| | | | | oxygenSaturation = poor: null
| | | bpStability = mod-stable: home
| | | bpStability = unstable
| | | | internalTemp = mid
| | | | | surfaceTempStability = stable: ward
| | | | | surfaceTempStability = mod-stable: null
| | | | | surfaceTempStability = unstable: home
| | | | internalTemp = high: ward
| | | | internalTemp = low: null
| | surfaceTemp = high
| | | bpStability = stable
| | | | internalTemp = mid: home
| | | | internalTemp = high: null
| | | | internalTemp = low: ward
| | | bpStability = mod-stable: ward
| | | bpStability = unstable: ward
| | surfaceTemp = low
| | | bpStability = stable: ward
| | | bpStability = mod-stable: home
| | | bpStability = unstable
| | | | internalTemp = mid: home
| | | | internalTemp = high: null
| | | | internalTemp = low: home
| bloodPressure = low: ward
comfort = 15
| bpStability = stable
| | bloodPressure = high: home
| | bloodPressure = mid
| | | internalTemp = mid: ward
| | | internalTemp = high: null
| | | internalTemp = low
| | | | surfaceTemp = mid: home
| | | | surfaceTemp = high: ward
| | | | surfaceTemp = low: null
| | bloodPressure = low: null
| bpStability = mod-stable: ward
| bpStability = unstable
| | surfaceTemp = mid: ward
| | surfaceTemp = high: ward
| | surfaceTemp = low: home
```

รูปที่ 4.1 โมเดลของข้อมูล breast-cancer recurrences สร้างจากโปรแกรม ID3

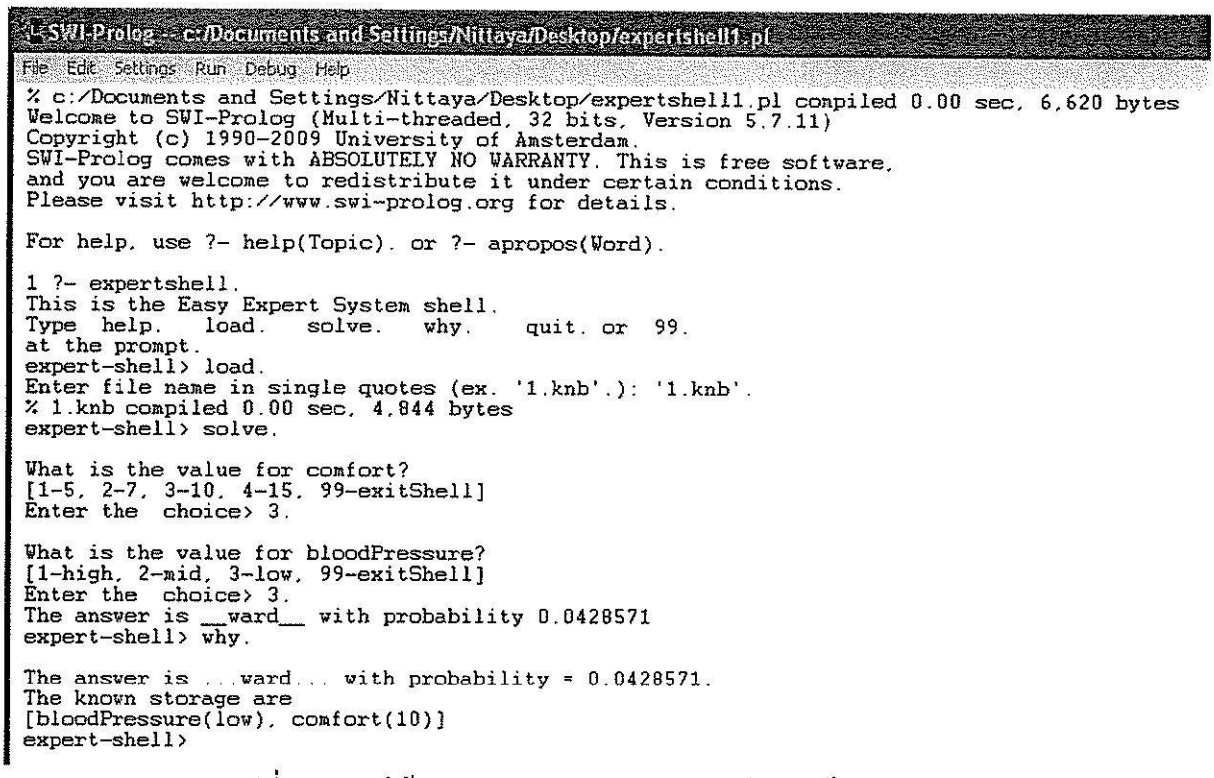
```
Sigmoid Node 0
   Inputs   Weights
   Threshold   4.838013774039095
   Node 2   -3.908582636087111
   Node 3   -3.162267392436414
   Node 4   1.0981619846468047
   Node 5   0.18908775947239695
   Node 6   -5.74938601906933
   Node 7   4.422399319810025
   Node 8   0.1999742473645021
   Node 9   -3.5324578914467377
   Node 10   0.11046630190222735
   Node 11   -1.7079259280625576
   Node 12   2.566868144965138
   Node 13   -1.1374211603297253
   Node 14   -2.7865026549167866
   Node 15   0.5042240856020124
Sigmoid Node 1
   Inputs   Weights
   Threshold   -4.837381256263012
   Node 2   3.9522624615165793
   Node 3   3.1682972818602746
   Node 4   -1.094499095299558
   Node 5   -0.13283814684003517
   Node 6   5.746872610366041
   Node 7   -4.424392875653778
   Node 8   -0.18722208735744159
   Node 9   3.527387810727785
   Node 10   -0.10472586398059963
   Node 11   1.6962159718097416
   Node 12   -2.560804638307743
   Node 13   1.0737048825607642
   Node 14   2.7877184996555053
   Node 15   -0.5273986540734854
Sigmoid Node 2
   Inputs   Weights
   Threshold   -0.27814804385505276
   Attrib internalTemp=mid   0.5810283641710741
   Attrib internalTemp=high   0.21179494161259477
   Attrib internalTemp=low   -0.5449481721523186
   Attrib surfaceTemp=mid   1.1141644074041142
   Attrib surfaceTemp=high   1.5150741028274102
   ...
...
...
Sigmoid Node 15
   Inputs   Weights
   Threshold   -0.2930395536777638
   Attrib internalTemp=mid   0.062120538342274045
   Attrib internalTemp=high   0.3128132567811115
   Attrib internalTemp=low   0.018576988447827396
   Attrib surfaceTemp=mid   0.04411226345984964
   Attrib surfaceTemp=high   0.5845640457361527
   Attrib surfaceTemp=low   -0.3416470192893855
   Attrib oxygenSaturation=excellent   -0.02240743383748446
   Attrib oxygenSaturation=good   -0.0017430323732858114
   Attrib oxygenSaturation=fair   -0.01474001719084124
   Attrib oxygenSaturation=poor   -0.02503529326219822
   Attrib bloodPressure=high   0.3330125093594764
   Attrib bloodPressure=mid   -0.3816134197463625
   Attrib bloodPressure=low   0.32842425053023594
   Attrib surfaceTempStability=stable   0.11418920446981239
   Attrib surfaceTempStability=mod-stable   -0.013239752658
   Attrib surfaceTempStability=unstable   -0.07986194908507
   Attrib coreTempStability=stable   -0.14532864396285983
   Attrib coreTempStability=mod-stable   0.3147436014254902
   Attrib coreTempStability=unstable   0.12862619667684774
   Attrib bpStability=stable   0.21950090896499233
   Attrib bpStability=mod-stable   -0.10037656428787754
   Attrib bpStability=unstable   0.1820629305281999
   Attrib comfort=05   0.04690470481373389
   Attrib comfort=07   0.09299279029838585
   Attrib comfort=10   -0.024417921014600892
   Attrib comfort=15   0.4216833375087676
Class home
   Input
   Node 0
Class ward
   Input
   Node 1
```

รูปที่ 4.2 โมเดลของข้อมูล breast-cancer recurrences สร้างจากโปรแกรม Multi-layer perceptron

โครงการวิจัยเรื่องการประมวลผลหลังกระบวนการทำเหมืองข้อมูลนี้ จึงมีวัตถุประสงค์หลักเพื่อจะพัฒนาโปรแกรมเพื่อรับอินพุตเป็นโมเดลข้อมูล จากนั้นแปลงโมเดลข้อมูลให้อยู่ในรูปแบบที่นักวิเคราะห์ข้อมูลโดยทั่วไปเข้าใจได้ง่ายกว่าในรูปแบบของ decision tree ในรูปที่ 4.1 หรือรูปแบบ sigmoid functions ในรูปที่ 4.2 นอกจากแปลงรูปแบบโมเดลข้อมูลให้เข้าใจได้ง่ายแล้ว งานวิจัยนี้ยังได้เพิ่มเทคนิคของการกลั่นกรองโมเดลให้มีขนาดเล็กลง ทั้งนี้เพื่อความสะดวกในการแปลผลและนำไปใช้งาน งานวิจัยนี้ได้อำนวยความสะดวกด้านการใช้งานโมเดลที่ได้รับการกลั่นกรองแล้วด้วยการจัดสร้างฐานความรู้โดยอัตโนมัติจากโมเดลข้อมูล และสร้าง expert system shell ที่ผู้ใช้สามารถสอบถามโมเดลข้อมูลที่อยู่ในฐานความรู้ได้อย่างสะดวก ตัวอย่างของฐานความรู้ที่สร้างจากข้อมูล post-operative patients และกลั่นกรองไว้เฉพาะโมเดลที่มีค่าความน่าจะเป็นสูงกว่า 0.05 แสดงได้ดังรูปที่ 4.3 และแสดงการใช้ experts system shell สอบถามโมเดลข้อมูลนี้ในรูปที่ 4.4

```
type(ward,0.1) :- comfort(10), bloodPressure(high), surfaceTemp(low).
type(ward,0.07142) :- comfort(10), bloodPressure(mid), surfaceTemp(low), bpStability(stable).
type(ward,0.07142) :- comfort(10), bloodPressure(mid), surfaceTemp(high), bpStability(mod_stable).
type(ward,0.05714) :- comfort(10), bloodPressure(high), surfaceTemp(mid), bpStability(mod_stable).
type(ward,0.04285) :- comfort(15), bpStability(unstable), surfaceTemp(mid).
type(ward,0.04285) :- comfort(10), bloodPressure(mid), surfaceTemp(mid), bpStability(stable),
                      internalTemp(mid), tempStability(unstable).
type(ward,0.04285) :- comfort(10), bloodPressure(low).
type(ward,0.02857) :- comfort(15), bpStability(unstable), surfaceTemp(high).
type(ward,0.02857) :- comfort(15), bpStability(stable), internalTemp(mid), surfaceTemp(mid).
type(ward,0.02857) :- comfort(15), bpStability(mod_stable).
type(home,0.02857) :- comfort(10), bloodPressure(mid), surfaceTemp(mid), bpStability(mod_stable).
type(ward,0.02857) :- comfort(10), bloodPressure(high), surfaceTemp(mid),
                      bpStability(stable), oxygenSaturation(excellent).
```

รูปที่ 4.3 โมเดลของข้อมูล post-operative patents ที่มีค่าความน่าจะเป็นสูงกว่า 0.02



รูปที่ 4.4 การใช้ expert system shell สอบถามโมเดลข้อมูล

ข้อเสนอแนะ

(1) โปรแกรมสร้างโมเดลข้อมูลที่พัฒนาขึ้นนี้มีข้อกำหนดเกี่ยวกับชนิดข้อมูล โดยข้อมูลที่ใช้จะต้องเป็นชนิด nominal หรือ categorical ถ้าข้อมูลใดเป็นจำนวนเลข (numeric) จะต้องแปลงตัวเลขให้เป็นข้อความ และการจำแนกคลาสของข้อมูลเป็นชนิด binary classification ดังนั้นแนวทางการปรับปรุงงานวิจัยนี้ต่อไปคือพัฒนาวิธีการจำแนกข้อมูลให้เป็น multi-class classification และเพิ่มฟังก์ชันการแปลงชนิดข้อมูลจากจำนวนเลขเป็นช่วงของค่า (interval)

(2) การสร้างกฎการตัดสินใจในงานวิจัยนี้สร้างจาก classification rules ทั้งนี้โมเดลที่ได้การทำเหมืองข้อมูลประเภทอื่น เช่น clustering model, association model สามารถแปลงให้เป็นกฎการตัดสินใจได้เช่นเดียวกัน

(3) จากแนวทางการลดขนาดของโมเดล ด้วยการใช้เฉพาะกฎการตัดสินใจที่มีค่าความน่าจะเป็นสูง จะทำให้ได้โมเดลที่มีจำนวนกฎน้อย แต่ผลที่อาจจะตามมาคือกฎที่ถูกคัดเลือกไว้อาจจะมีเงื่อนไขการพิจารณาไม่ครบถ้วนทำให้โมเดลไม่สมบูรณ์ หรือกฎบางส่วนอาจขัดแย้งกัน แนวทางแก้ไขปัญหากรณีเงื่อนไขไม่ครบถ้วนคือจะต้องมี default rules เตรียมไว้ ส่วนในกรณีที่กฎขัดแย้งอาจใช้วิธีการโหวตจากทุกกฎที่เกี่ยวข้อง จากนั้นใช้ค่าทำนายเป็นค่าส่วนใหญ่จากผลโหวต

(4) ในงานทำเหมืองข้อมูล บางครั้งผู้วิเคราะห์ข้อมูลสนใจในกฎที่มีค่าความน่าจะเป็นต่ำ (นั่นคือ มีโอกาสเกิดข้อมูลนี้น้อยกว่าข้อมูลอื่นๆ) แต่เป็นกฎที่มีความถูกต้องสูง ซึ่งเรียกข้อมูลในลักษณะนี้ว่า rare cases การค้นหากฎในลักษณะเช่นนี้ทำได้โดยการเรียกใช้โปรแกรม แล้วกำหนดเกณฑ์ความน่าจะเป็นขั้นต่ำที่ 0.0 (นั่นคือให้โปรแกรมค้นหาโมเดลในลักษณะของกฎ โดยค้นหาทุกกฎที่เป็นไปได้) แต่ผลข้างเคียงของการกำหนดค่าเช่นนี้อาจจะต้องใช้เนื้อที่หน่วยความจำมากขึ้นในการประมวลผลโปรแกรม และโปรแกรมอาจใช้เวลาประมวลผลนานขึ้น

(5) การทดสอบโปรแกรมในงานวิจัยนี้ใช้ข้อมูลทดสอบจากแหล่งข้อมูลมาตรฐาน UCI ซึ่งถึงแม้ในการทดลองจะให้ผลที่ดี แต่ในข้อมูลที่เกิดขึ้นจริงบางครั้งการกระจายของข้อมูลไม่เป็นรูปแบบปกติ หรือเป็นข้อมูลที่กระจัดกระจายไม่มีรูปแบบที่ชัดเจนทำให้ไม่ปรากฏโมเดล ดังนั้นในการนำโปรแกรมนี้ไปใช้งานจริงจึงต้องมีการทดสอบเบื้องต้นกับชุดข้อมูลขนาดเล็ก และถ้าโมเดลมีคุณภาพต่ำเกินไป อาจต้องพิจารณาปรับปรุงคุณภาพข้อมูลโดยอาจจะตัดบางแอททริบิวต์ออก เพิ่มบางแอททริบิวต์ หรือเพิ่มจำนวนข้อมูล

(6) โปรแกรมที่พัฒนาขึ้นนี้มีลักษณะเป็น rapid prototyping ทำให้ส่วนติดต่อกับผู้ใช้ยังไม่ดีเท่าที่ควร ดังนั้นแนวทางการปรับปรุงงานวิจัยนี้จึงรวมการพัฒนา GUI (graphical user interface) เพื่อให้โปรแกรมใช้งานได้ง่ายขึ้น

บรรณานุกรม

G. Adomavicius and A. Tuzhilin. Expert-driven validation of rule-based user models in personalization applications. *Journal of Data Mining and Knowledge Discovery*, 5(1/2): 33-58, 2001.

R.J. Bayardo, R. Agrawal, and D. Gunopulos. Constraint-based rule mining in large, dense databases. In *Proceedings of the 15th International Conference on Data Engineering*, March 1999.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and regression tree.* Belmont: Wadsworth, 1984.

S. Brin, R. Motwani, J.D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the ACM SIGMOD Conference*, 1997.

P. Clark and S. Matwin. Using qualitative models to guide induction learning. In *Proceedings of the International Conference on Machine Learning*, 1993.

W.J. Frawley, G. Piatetesky-Shapiro, and C.J. Matheus. Knowledge discovery in databases: an overview. In G.Piatetsky-Shapiro and W.J. Frawley (Eds.), *Knowledge Discovery in Databases*, AAAI/MIT Press, 1991.

M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A.I. Verkamo. Finding interesting rules from large sets of discovered association rules. In *Proceedings of the Third International Conference on Information and Knowledge Management*, December 1994.

J. Major and J. Mangano. Selecting among rules induced from a hurricane database. In *Proceedings of the AAAI-93 Workshop on KDD*, 1993.

J. Ortega and D. Fisher. Flexibly exploiting prior knowledge in empirical learning. *IJCAI*, 1995.

M. Pazzani and D. Kibler. The utility of knowledge in inductive learning. *Machine Learning*, 9, 1992.

G. Piatetsky-Shapiro and C.J. Matheus. The interestingness of deviations. In *Proceedings of the AAAI-94 Workshop on Knowledge Discovery in Databases*, 1994.

G. Piatetsky-Shapiro, C.J. Matheus, P. Smyth, and R. Uthurusamy. KDD-93:progress and challenges. *AI Magazine*, Fall, 77-87, 1994.

J.R. Quinlan. *C4.5: Program for Machine Learning*. Morgan Kaufmann, 1992.

A. Silberschatz and A. Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, Montreal, Canada, August 1995.

A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(6), December 1996.

E. Suzuki. Autonomous discovery of reliable exception rules. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, August 1997.

K.Wang, S.H.W. Tay, and B. Liu. Interestingness-based interval merger for numeric association rules. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, August 1998.

ภาคผนวก

ภาคผนวก ก

รหัสต้นฉบับของโปรแกรมการประมวลผลหลังกระบวนการทำเหมืองข้อมูล

```
/* ============  Post-Data Mining : main program ========= */

/*-------- How to run  ------
1. call  id3menuAll3.pl  --> expertshell1.pl
2. expertshell.
3. load.      and input '1.knb'.
4. solve.
5. why.
6. quit.
----------------------------*/

:-dynamic known/1, answer/2.

expertshell :-
            greeting,
            repeat,
            write('expert-shell> '),
            read(X),
            do(X),
            (X == quit;X == 99),
            writeln('>>>>Goodbye, see you later<<<<'), !.

greeting :-
            write('This is the Easy Expert System shell.'), nl,
            native_help.

do(help) :- native_help, !.
do(load) :- load_kb, !.
do(solve) :- solve, !.
do(why):-why,!.
do(quit). do(99).

do(X) :-    write(X),
            write(' is not a legal command.'), nl, fail.

native_help :-
            write('Type  help.   load.   solve.   why.    quit. or  99.'),nl,
            write('at the prompt.'), nl.

load_kb :-  write('Enter file name in single quotes (ex. ''1.knb''.): '),
            read(F),
            reconsult(F).

solve :-    retractall(known( _) ),retractall(answer(_,_)),
            top_goal(X,V),
            format('The answer is __~w__ with probability ~w',[X,V]),
            assert(answer(X,V)),nl.

solve :-    write('No answer found.'),nl.

menuask(Pred,Value,Menu):-
            menuask(Pred,Menu),
            atomic_list_concat([Pred,'(',Value,')'],X),
            term_to_atom(T,X),known(T),!.

menuask(Pred,_):-
       atomic_list_concat([Pred,'(','_',')'],X),  % check for recorded predicate
       term_to_atom(T,X),known(T),!.           % not ask again

menuask(Attribute,Menu):-
            nl,write('What is the value for '),
            write(Attribute),write('?'),nl,
            addchoice(Menu,MenuRes),writeln(MenuRes),
```

```
            write('Enter the  choice> '),
            read(C),
            member(C-V,MenuRes),
            (C=99 -> abort ; true),
            atomic_list_concat([Attribute,'(',V,')'],X),
            term_to_atom(T,X),
            asserta(known(T)).

why:- answer(A,V),
      format('~nThe answer is ...~w... with probability = ~w.~n',[A,V]),
      findall( X , known(X),Result),
      writeln('The known storage are'),
      writeln(Result).

addchoice(X,Res):-
       length(X,Len),
       numlist(1,Len,NumL),
       map(NumL,X,Res).

map([],[],[99-exitShell]).
map([H|T],[X|TT],[H-X|T1]):- map(T,TT,T1).

% ========= End of Expert System Shell Program ===========



%=====================================================
%  Rule induction program
%    ID3-based algorithm with probabilistic values
%=====================================================

addAllKnowledge:-
       findall([A],pathFromRootToLeaf(A,_),Res)       ,
       retractall(_>>_>>_),
       maplist(apply(assert),Res),
       write(addToKNB),nl.  % add to knowledge base

selectRule(V,Res):-
       findall(N>>X>>Class,(X>>Class>>N,N>=V),Res1),
       sort(Res1,Res2),
       reverse(Res2,Res).

path(A,[H|T],C):-
       edge(A,H,B),
       path(B,T,C).
path(C,[],C):-!.

pathFromRootToLeaf(V>>Class>>Num,C):-
       path(1,V,C),
       node(C,Value1-Value2),
       (Value1=[] ; Value2=[]),
       (Value1=[]->length(Value2,Numb) ; length(Value1,Numb)),
       total+Total,
       Num is Numb/Total,hasClass(C1,C2),
       (Value1=[]->Class=C2 ; Class=C1).

min_cand([H|T], Min) :-
       min_cand(T, H, Min).
min_cand([], Min, Min).
min_cand([H|T], Min0, Min) :-
       H=[V,_,_],Min0=[V0,_,_],
       ( V<V0 ->Min1=H;Min1=Min0),
       min_cand(T, Min1, Min).
```

```
cand_node([H|T],CurInstL,[[Val,H,SpliteL]|OtherAttr]):-
        info(H,CurInstL,Val,SpliteL),
        cand_node(T,CurInstL,OtherAttr).

cand_node([],_,[]):-!.
cand_node(_,[],[]).

concat3(A,B,C,R):-
        atom_concat(A,B,R1),
        atom_concat(R1,C,R).

info(A,CurInstL,R,Splite):-
        attribute(A,L),
        maplist(concat3(A,=),L,L1),%make a good form
        suminfo(L1,CurInstL,R,Splite).

suminfo([H|T],CurInstL,R,[Splite|ST]) :-
        AllBag=CurInstL,hasClass(C1,C2),
        term_to_atom(H1,H),
        findall(X1,(instance(X1,_,L1),member(X1,CurInstL),
                    member(H1,L1)),BagGro),
        findall(X2,(instance(X2,class=C1,L2),member(X2,CurInstL),
                    member(H1,L2)),BagPos),
        findall(X3,(instance(X3,class=C2,L3),member(X3,CurInstL),
                    member(H1,L3)),BagNeg),
        (H11=H22) =H1,
        length(AllBag,Nall),
        length(BagGro,NGro),
        length(BagPos,NPos),
        length(BagNeg,NNeg),
        Splite=H11-H22/BagPos-BagNeg,
        suminfo(T,CurInstL,R1,ST),
        (   NPos is 0 *->L1 = 0; L1 is (log(NPos/NGro)/log(2)) ),
        (   0 is NNeg *->L2 = 0; L2 is (log(NNeg/NGro)/log(2)) ),
        (   NGro is 0 -> R= 999;
              R is (NGro/Nall)*(-(NPos/NGro)*L1-(NNeg/NGro)*L2)+R1 ) .
suminfo([],_,0,[]).

%-------------- ID3 ------------------

:- dynamic current_node/1,node/2,edge/3,hasClass/2,type/2.

init(AllAttr,[root-nil/PB-NB]):-
       retractall(hasClass(_,_)),
       attribute( class,[ Y1, Y2]),
       assert(hasClass(Y1,Y2)),
       retractall(node(_,_)),
       retractall(current_node(_)),
       retractall(type(_,_)),
       retractall(edge(_,_,_)),
       assert(current_node(0)) ,
       hasClass(C1,C2),
       findall(X,attribute(X,_),AllAttr1),delete(AllAttr1,class,AllAttr),
       findall(X2,instance(X2,class=C1,_),PB),
       findall(X3,instance(X3,class=C2,_),NB),
       length(PB,N1),length(NB,N2),N is N1+N2,
       retractall(total+_),
       apply(assert,[total+N]).
```

```
getnode(X):-
       current_node(X),
       X1 is X+1,
       retractall(current_node(_)),
       assert(current_node(X1)),
       X1 <4000.       % limit tree size at 4000 nodes

create_edge_onelevel(_,_,[]):-!.
create_edge_onelevel(_,[],_):-!.
create_edge_onelevel(N,AllAttr,EdgeList):- create_nodes(N,AllAttr,EdgeList).

create_nodes(N,AllAttr,[H1-H2/PB-NB|T]):-
       getnode(N1),
       assert(edge(N,H1=H2,N1)),
       assert(node(N1,PB-NB)),
       append(PB,NB,AllInst),
       ( (PB\==[], NB\==[])->
             (cand_node(AllAttr,AllInst,AllSplite),
              min_cand(AllSplite,[V,MinAttr,Splite]),
              delete(AllAttr,MinAttr,Attr2),
              create_edge_onelevel( N1,Attr2,Splite)) ; true ),
       create_nodes(N,AllAttr,T).

create_nodes(_,_,[]):-!.
create_nodes(_,[],_):-!.

mainId3(Min):-
       init(AllAttr,EdgeList),
       getnode(N),
       create_edge_onelevel(N,AllAttr,EdgeList),
       addAllKnowledge,
       selectRule(Min,Res),
       writeln(Res),
       tell('1.knb'),
       writeHeadF,
       maplist(createRule1,Res),
       nl,writeTailF,
       told, writeln(endProcess).

%------------------------
%  Generate rules in KB
%------------------------
writeHeadF :-
  format('% 1.knb ~n% for expert shell. --- written by Postprocess'),
  format('~n% top_goal where the inference starts.~n'),
  format('~ntop_goal(X,V) :- type(X,V).~n').

writeTailF:-
  findall(_,(attribute(S,L),
  format('~n~w(X):-menuask(~w,X,~w). %generated menu',[S,S,L])),_),
  format('~n~n%end of automatic post process').

%--------------  MENU ----------------
id3menu:-
  new(Dialog,dialog('Create Rules from Id3')),
  send_list(Dialog, append,
  [ new(D1, text_item(datafile,'post-operative.pl')),
    new(Per,text_item(minProb,'0.016')),
    button(cancel, message(Dialog, destroy)),
    button(enter, and(message(@prolog,callId3,D1?selection,Per?selection ),
    message(Dialog, destroy)))  % enter&destroy
  ]),
  send(Dialog, open).
```

```
%
callId3(Dfile,Per):-
       term_to_atom(Per1,Per),
       consult(Dfile),
       mainId3(Per1).

:-id3menu.

%---------------

transform1([X=V],[Res]):-
       atomic_list_concat([X,'(',V,')'],Res1),
       term_to_atom(Res,Res1),!.

transform1([X=V|T],[Res|T1]):-
       atomic_list_concat([X,'(',V,')'],Res1),
       term_to_atom(Res,Res1),
       transform1(T,T1).

createRule1(I):-
       I=Z>>X>>Y,
       transform1(X,BodyL),
       format('~ntype(~w,~w):-',[Y,Z]),
       myformat(BodyL) ,
       write('  % generated rule'),!.

myformat([X]):-write(X),write('.'),!.
myformat([H|T]):-write(H),write(','),myformat(T).


% ================== End of Rule Induction Program =======
```

```
% ------------------------------------
%      Data File:  Post-operative.pl
% ------------------------------------
%   class home = after operation patient prepared to go home
%   class ward = patient not in good condition: sent to general floor

attribute( internalTemp,       [mid, high, low] ).
attribute( surfaceTemp,        [mid,high, low] ).
attribute( oxygenSaturation,   [excellent, good, fair, poor] ).
attribute( bloodPressure,      [high, mid, low] ).
attribute( tempStability,      [stable, mod_stable, unstable] ).
attribute( coreTempStability,  [stable,mod_stable, unstable] ).
attribute( bpStability,        [stable, mod_stable, unstable] ).
attribute( comfort,            [5, 7, 10, 15] ).
attribute( class,              [ home, ward]).

instance(1, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=excellent,
                         bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                         bpStability=stable, comfort=15] ).
instance(2, class=home, [internalTemp=mid, surfaceTemp=high, oxygenSaturation=excellent,
                         bloodPressure=high, tempStability=stable, coreTempStability=stable,
                         bpStability=stable, comfort=10] ).
instance(3, class=ward, [internalTemp=high, surfaceTemp=low, oxygenSaturation=excellent,
                         bloodPressure=high, tempStability=stable, coreTempStability=stable,
                         bpStability=mod_stable, comfort=10] ).
instance(4, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=good,
                         bloodPressure=high, tempStability=stable, coreTempStability=unstable,
                         bpStability=mod_stable, comfort=15] ).
instance(5, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                         bloodPressure=high, tempStability=stable, coreTempStability=stable,
                         bpStability=stable, comfort=10] ).
instance(6, class=home, [internalTemp=high, surfaceTemp=low, oxygenSaturation=good,
                         bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                         bpStability=unstable, comfort=15] ).
instance(7, class=home, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=excellent,
                         bloodPressure=high, tempStability=stable, coreTempStability=stable,
                         bpStability=mod_stable, comfort=5] ).
instance(8, class=home, [internalTemp=high, surfaceTemp=mid, oxygenSaturation=excellent,
                         bloodPressure=mid, tempStability=unstable,
                         coreTempStability=unstable, bpStability=stable, comfort=10]).
instance(9, class=home, [internalTemp=mid, surfaceTemp=high, oxygenSaturation=good,
                         bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                         bpStability=stable, comfort=10] ).
instance(10, class=home, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=excellent,
                         bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                         bpStability=mod_stable, comfort=10] ).
instance(11, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                         bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                         bpStability=stable, comfort=15]).
instance(12, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=good,
                         bloodPressure=high, tempStability=stable, coreTempStability=stable,
                         bpStability=mod_stable, comfort=10] ).
instance(13, class=ward, [internalTemp=high, surfaceTemp=high, oxygenSaturation=excellent,
                         bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                         bpStability=unstable, comfort=15] ).
instance(14, class=ward, [internalTemp=mid, surfaceTemp=high, oxygenSaturation=good,
                         bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                         bpStability=mod_stable, comfort=10] ).
instance(15, class=home, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=good,
                         bloodPressure=high, tempStability=unstable,
                         coreTempStability=unstable, bpStability=stable, comfort=15]).
instance(16, class=ward, [internalTemp=high, surfaceTemp=high, oxygenSaturation=excellent,
                         bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                         bpStability=unstable, comfort=10] ).
instance(17, class=ward, [internalTemp=low, surfaceTemp=high, oxygenSaturation=good,
                         bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                         bpStability=mod_stable, comfort=15]).
instance(18, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=good,
                         bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                         bpStability=stable, comfort=10] ).
```

```
instance(19, class=ward, [internalTemp=mid, surfaceTemp=high, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=15] ).
instance(20, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10] ).
instance(21, class=ward, [internalTemp=low, surfaceTemp=high, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=15] ).
instance(22, class=home, [internalTemp=low, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=10] ).
instance(23, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=15] ).
instance(24, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10] ).
instance(25, class=ward, [internalTemp=high, surfaceTemp=high, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10] ).
instance(26, class=ward, [internalTemp=low, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10] ).
instance(27, class=ward, [internalTemp=high, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=low, tempStability=stable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10] ).
instance(28, class=ward, [internalTemp=low, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10] ).
instance(29, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=unstable, comfort=15]).
instance(30, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(31, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(32, class=ward, [internalTemp=low, surfaceTemp=low, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(33, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(34, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(35, class=ward, [internalTemp=low, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(36, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(37, class=home, [internalTemp=low, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(38, class=home, [internalTemp=low, surfaceTemp=low, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(39, class=home, [internalTemp=low, surfaceTemp=low, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=7]).
instance(40, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(41, class=ward, [internalTemp=low, surfaceTemp=low, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
```

```
instance(42, class=home, [internalTemp=low, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=15]).
instance(43, class=home, [internalTemp=high, surfaceTemp=high, oxygenSaturation=good,
                          bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=15]).
instance(44, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(45, class=ward, [internalTemp=low, surfaceTemp=low, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(46, class=home, [internalTemp=low, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(47, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(48, class=ward, [internalTemp=mid, surfaceTemp=high, oxygenSaturation=good,
                          bloodPressure=low, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(49, class=ward, [internalTemp=mid, surfaceTemp=high, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable,
                          coreTempStability=mod_stable, bpStability=mod_stable, comfort=10]).
instance(50, class=ward, [internalTemp=low, surfaceTemp=high, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(51, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(52, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(53, class=ward, [internalTemp=high, surfaceTemp=high, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(54, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=15]).
instance(55, class=ward, [internalTemp=high, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=unstable, comfort=15]).
instance(56, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=good,
                          bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(57, class=ward, [internalTemp=low, surfaceTemp=low, oxygenSaturation=good,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(58, class=ward, [internalTemp=mid, surfaceTemp=high, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(59, class=ward, [internalTemp=mid, surfaceTemp=high, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(60, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(61, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(62, class=home, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(63, class=ward, [internalTemp=high, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable,
                          coreTempStability=unstable, bpStability=unstable, comfort=10]).
instance(64, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
```

```
instance(65, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(66, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(67, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(68, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=low, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(69, class=ward, [internalTemp=low, surfaceTemp=low, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(70, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
%
/* *** Test Data
%     ==========
instance(71, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(72, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(73, class=ward, [internalTemp=low, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(74, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(75, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(76, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable,
                          coreTempStability=unstable, bpStability=stable, comfort=10]).
instance(77, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(78, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=15]).
instance(79, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(80, class=ward, [internalTemp=high, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=5]).
instance(81, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(82, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(83, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=15]).
instance(84, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=15]).
instance(85, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(86, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=15]).
*/
```

```
% ------------------------------
%          Data: Breast-cancer.pl
% ------------------------------
%   Diagnosis recurrence of breast cancer
%   class = yes : recurrent events
%   class = no  : no recurrence
%
attribute(age,       [range20_29, range30_39, range40_49, range50_59, range60_69]).
attribute(menopause, [lt40, ge40, premeno]).
attribute(tumorSize, [rang0_4, range5_9, range10_14, range15_19, range20_24, range25_29,
                      range30_34, range35_39, range40_44, range45_49, range50_54]).
attribute(invNodes, [range0_2, range3_5, range6_8, range9_11, range15_17,  range24_26]).
attribute(nodeCaps, [missing, yes, no]).
attribute(degMalig, [1, 2, 3]).
attribute(breast,   [left, right]).
attribute(breastQuad, [left_up, left_low, right_up, right_low, central]).
attribute(irradiat, [yes, no]).
attribute(class,    [no, yes]).

instance(1, class=yes, [age=range40_49, menopause=premeno, tumorSize=range15_19,
                        invNodes=range0_2, nodeCaps=yes, degMalig=3, breast=right,
                        breastQuad=left_up, irradiat=no]).
instance(2, class=no, [age=range50_59, menopause=ge40, tumorSize=range15_19,
                        invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                        breastQuad=central, irradiat=no]).
instance(3, class=yes, [age=range50_59, menopause=ge40, tumorSize=range35_39,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                        breastQuad=left_low, irradiat=no]).
instance(4, class=no, [age=range40_49, menopause=premeno, tumorSize=range35_39,
                        invNodes=range0_2, nodeCaps=yes, degMalig=3, breast=right,
                        breastQuad=left_low, irradiat=yes]).
instance(5, class=yes, [age=range40_49, menopause=premeno, tumorSize=range30_34,
                        invNodes=range3_5, nodeCaps=yes, degMalig=2, breast=left,
                        breastQuad=right_up, irradiat=no]).
instance(6, class=no, [age=range50_59, menopause=premeno, tumorSize=range25_29,
                        invNodes=range3_5, nodeCaps=no, degMalig=2, breast=right,
                        breastQuad=left_up, irradiat=yes]).
instance(7, class=no, [age=range50_59, menopause=ge40, tumorSize=range40_44,
                        invNodes=range0_2, nodeCaps=no, degMalig=3, breast=left,
                        breastQuad=left_up, irradiat=no]).
instance(8, class=no, [age=range40_49, menopause=premeno, tumorSize=range10_14,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                        breastQuad=left_up, irradiat=no]).
instance(9, class=no, [age=range40_49, menopause=premeno, tumorSize=range0_4,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                        breastQuad=right_low, irradiat=no]).
instance(10, class=no, [age=range40_49, menopause=ge40, tumorSize=range40_44,
                        invNodes=range15_17, nodeCaps=yes, degMalig=2, breast=right,
                        breastQuad=left_up, irradiat=no]).
instance(11, class=no, [age=range50_59, menopause=premeno, tumorSize=range25_29,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                        breastQuad=left_low, irradiat=no]).
instance(12, class=no, [age=range60_69, menopause=ge40, tumorSize=range15_19,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                        breastQuad=left_up, irradiat=no]).
instance(13, class=no, [age=range50_59, menopause=ge40, tumorSize=range30_34,
                        invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                        breastQuad=central, irradiat=no]).
instance(14, class=no, [age=range50_59, menopause=ge40, tumorSize=range25_29,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                        breastQuad=left_up, irradiat=no]).
instance(15, class=yes, [age=range40_49, menopause=premeno, tumorSize=range25_29,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                        breastQuad=left_low, irradiat=yes]).
instance(16, class=no, [age=range30_39, menopause=premeno, tumorSize=range20_24,
                        invNodes=range0_2, nodeCaps=no, degMalig=3, breast=left,
                        breastQuad=central, irradiat=no]).
instance(17, class=no, [age=range50_59, menopause=premeno, tumorSize=range10_14,
                        invNodes=range3_5, nodeCaps=no, degMalig=1, breast=right,
                        breastQuad=left_up, irradiat=no]).
```

```
instance(18, class=no, [age=range60_69, menopause=ge40, tumorSize=range15_19,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                        breastQuad=left_up, irradiat=no]).
instance(19, class=no, [age=range50_59, menopause=premeno, tumorSize=range40_44,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                        breastQuad=left_up, irradiat=no]).
instance(20, class=no, [age=range50_59, menopause=ge40, tumorSize=range20_24,
                        invNodes=range0_2, nodeCaps=no, degMalig=3, breast=left,
                        breastQuad=left_up, irradiat=no]).
instance(21, class=yes, [age=range50_59, menopause=lt40, tumorSize=range20_24,
                        invNodes=range0_2, nodeCaps=missing, degMalig=1, breast=left,
                        breastQuad=left_low, irradiat=no]).
instance(22, class=no, [age=range60_69, menopause=ge40, tumorSize=range40_44,
                        invNodes=range3_5, nodeCaps=no, degMalig=2, breast=right,
                        breastQuad=left_up, irradiat=yes]).
instance(23, class=no, [age=range50_59, menopause=ge40, tumorSize=range15_19,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                        breastQuad=left_low, irradiat=no]).
instance(24, class=no, [age=range40_49, menopause=premeno, tumorSize=range10_14,
                        invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                        breastQuad=left_up, irradiat=no]).
instance(25, class=yes, [age=range30_39, menopause=premeno, tumorSize=range15_19,
                        invNodes=range6_8, nodeCaps=yes, degMalig=3, breast=left,
                        breastQuad=left_low, irradiat=yes]).
instance(26, class=no, [age=range50_59, menopause=ge40, tumorSize=range20_24,
                        invNodes=range3_5, nodeCaps=yes, degMalig=2, breast=right,
                        breastQuad=left_up, irradiat=no]).
instance(27, class=no, [age=range50_59, menopause=ge40, tumorSize=range10_14,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                        breastQuad=left_low, irradiat=no]).
instance(28, class=no, [age=range40_49, menopause=premeno, tumorSize=range10_14,
                        invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                        breastQuad=left_up, irradiat=no]).
instance(29, class=no, [age=range60_69, menopause=ge40, tumorSize=range30_34,
                        invNodes=range3_5, nodeCaps=yes, degMalig=3, breast=left,
                        breastQuad=left_low, irradiat=no]).
instance(30, class=yes, [age=range40_49, menopause=premeno, tumorSize=range15_19,
                        invNodes=range15_17, nodeCaps=yes, degMalig=3, breast=left,
                        breastQuad=left_low, irradiat=no]).
instance(31, class=yes, [age=range60_69, menopause=ge40, tumorSize=range30_34,
                        invNodes=range0_2, nodeCaps=no, degMalig=3, breast=right,
                        breastQuad=central, irradiat=no]).
instance(32, class=no, [age=range60_69, menopause=ge40, tumorSize=range25_29,
                        invNodes=range3_5, nodeCaps=missing, degMalig=1, breast=right,
                        breastQuad=left_low, irradiat=yes]).
instance(33, class=no, [age=range50_59, menopause=ge40, tumorSize=range25_29,
                        invNodes=range0_2, nodeCaps=no, degMalig=3, breast=left,
                        breastQuad=right_up, irradiat=no]).
instance(34, class=no, [age=range50_59, menopause=ge40, tumorSize=range20_24,
                        invNodes=range0_2, nodeCaps=no, degMalig=3, breast=right,
                        breastQuad=left_up, irradiat=no]).
instance(35, class=yes, [age=range40_49, menopause=premeno, tumorSize=range30_34,
                        invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                        breastQuad=left_low, irradiat=yes]).
instance(36, class=no, [age=range30_39, menopause=premeno, tumorSize=range15_19,
                        invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                        breastQuad=left_low, irradiat=no]).
instance(37, class=no, [age=range40_49, menopause=premeno, tumorSize=range10_14,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                        breastQuad=left_up, irradiat=no]).
instance(38, class=no, [age=range60_69, menopause=ge40, tumorSize=range45_49,
                        invNodes=range6_8, nodeCaps=yes, degMalig=3, breast=left,
                        breastQuad=central, irradiat=no]).
instance(39, class=no, [age=range40_49, menopause=ge40, tumorSize=range20_24,
                        invNodes=range0_2, nodeCaps=no, degMalig=3, breast=left,
                        breastQuad=left_low, irradiat=no]).
instance(40, class=no, [age=range40_49, menopause=premeno, tumorSize=range10_14,
                        invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                        breastQuad=right_low, irradiat=no]).
```

```
instance(41, class=yes, [age=range30_39, menopause=premeno, tumorSize=range35_39,
                         invNodes=range0_2, nodeCaps=no, degMalig=3, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(42, class=no, [age=range40_49, menopause=premeno, tumorSize=range35_39,
                         invNodes=range9_11, nodeCaps=yes, degMalig=2, breast=right,
                         breastQuad=right_up, irradiat=yes]).
instance(43, class=no, [age=range60_69, menopause=ge40, tumorSize=range25_29,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=left_low, irradiat=no]).
instance(44, class=yes, [age=range50_59, menopause=ge40, tumorSize=range20_24,
                         invNodes=range3_5, nodeCaps=yes, degMalig=3, breast=right,
                         breastQuad=right_up, irradiat=no]).
instance(45, class=no, [age=range30_39, menopause=premeno, tumorSize=range15_19,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(46, class=yes, [age=range50_59, menopause=premeno, tumorSize=range30_34,
                         invNodes=range0_2, nodeCaps=no, degMalig=3, breast=left,
                         breastQuad=right_up, irradiat=no]).
instance(47, class=no, [age=range60_69, menopause=ge40, tumorSize=range10_14,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=left_up, irradiat=yes]).
instance(48, class=no, [age=range40_49, menopause=premeno, tumorSize=range35_39,
                         invNodes=range0_2, nodeCaps=yes, degMalig=3, breast=right,
                         breastQuad=left_up, irradiat=yes]).
instance(49, class=no, [age=range50_59, menopause=premeno, tumorSize=range50_54,
                         invNodes=range0_2, nodeCaps=yes, degMalig=2, breast=right,
                         breastQuad=left_up, irradiat=yes]).
instance(50, class=no, [age=range50_59, menopause=ge40, tumorSize=range40_44,
                         invNodes=range0_2, nodeCaps=no, degMalig=3, breast=right,
                         breastQuad=left_up, irradiat=no]).
instance(51, class=yes, [age=range60_69, menopause=ge40, tumorSize=range15_19,
                         invNodes=range9_11, nodeCaps=missing, degMalig=1, breast=left,
                         breastQuad=left_low, irradiat=yes]).
instance(52, class=no, [age=range50_59, menopause=lt40, tumorSize=range30_34,
                         invNodes=range0_2, nodeCaps=no, degMalig=3, breast=right,
                         breastQuad=left_up, irradiat=no]).
instance(53, class=no, [age=range40_49, menopause=premeno, tumorSize=range0_4,
                         invNodes=range0_2, nodeCaps=no, degMalig=3, breast=left,
                         breastQuad=central, irradiat=no]).
instance(54, class=no, [age=range60_69, menopause=ge40, tumorSize=range40_44,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                         breastQuad=right_up, irradiat=no]).
instance(55, class=no, [age=range40_49, menopause=premeno, tumorSize=range25_29,
                         invNodes=range0_2, nodeCaps=missing, degMalig=2, breast=left,
                         breastQuad=right_low, irradiat=yes]).
instance(56, class=no, [age=range50_59, menopause=ge40, tumorSize=range25_29,
                         invNodes=range15_17, nodeCaps=yes, degMalig=3, breast=right,
                         breastQuad=left_up, irradiat=no]).
instance(57, class=no, [age=range50_59, menopause=premeno, tumorSize=range20_24,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(58, class=no, [age=range50_59, menopause=ge40, tumorSize=range35_39,
                         invNodes=range15_17, nodeCaps=no, degMalig=3, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(59, class=no, [age=range50_59, menopause=ge40, tumorSize=range50_54,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                         breastQuad=right_up, irradiat=no]).
instance(60, class=yes, [age=range30_39, menopause=premeno, tumorSize=range0_4,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=central, irradiat=no]).
instance(61, class=yes, [age=range50_59, menopause=ge40, tumorSize=range40_44,
                         invNodes=range6_8, nodeCaps=yes, degMalig=3, breast=left,
                         breastQuad=left_low, irradiat=yes]).
instance(62, class=no, [age=range40_49, menopause=premeno, tumorSize=range30_34,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=right_up, irradiat=yes]).
instance(63, class=no, [age=range40_49, menopause=ge40, tumorSize=range20_24,
                         invNodes=range0_2, nodeCaps=no, degMalig=3, breast=left,
                         breastQuad=left_up, irradiat=no]).
```

```
instance(64, class=yes, [age=range40_49, menopause=premeno, tumorSize=range30_34,
                         invNodes=range15_17, nodeCaps=yes, degMalig=3, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(65, class=yes, [age=range40_49, menopause=ge40, tumorSize=range20_24,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=left_up, irradiat=no]).
instance(66, class=no, [age=range50_59, menopause=ge40, tumorSize=range15_19,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                         breastQuad=central, irradiat=no]).
instance(67, class=no, [age=range30_39, menopause=premeno, tumorSize=range25_29,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=left_low, irradiat=no]).
instance(68, class=no, [age=range60_69, menopause=ge40, tumorSize=range15_19,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(69, class=yes, [age=range50_59, menopause=premeno, tumorSize=range50_54,
                         invNodes=range9_11, nodeCaps=yes, degMalig=2, breast=right,
                         breastQuad=left_up, irradiat=no]).
instance(70, class=no, [age=range30_39, menopause=premeno, tumorSize=range10_14,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                         breastQuad=left_low, irradiat=no]).
instance(71, class=yes, [age=range50_59, menopause=premeno, tumorSize=range25_29,
                         invNodes=range3_5, nodeCaps=yes, degMalig=3, breast=left,
                         breastQuad=left_low, irradiat=yes]).
instance(72, class=no, [age=range60_69, menopause=ge40, tumorSize=range25_29,
                         invNodes=range3_5, nodeCaps=missing, degMalig=1, breast=right,
                         breastQuad=left_up, irradiat=yes]).
instance(73, class=no, [age=range60_69, menopause=ge40, tumorSize=range10_14,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                         breastQuad=left_low, irradiat=no]).
instance(74, class=yes, [age=range50_59, menopause=ge40, tumorSize=range30_34,
                         invNodes=range6_8, nodeCaps=yes, degMalig=3, breast=left,
                         breastQuad=right_low, irradiat=no]).
instance(75, class=yes, [age=range30_39, menopause=premeno, tumorSize=range25_29,
                         invNodes=range6_8, nodeCaps=yes, degMalig=3, breast=left,
                         breastQuad=right_low, irradiat=yes]).
instance(76, class=no, [age=range50_59, menopause=ge40, tumorSize=range10_14,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(77, class=no, [age=range50_59, menopause=premeno, tumorSize=range15_19,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(78, class=no, [age=range40_49, menopause=premeno, tumorSize=range25_29,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=central, irradiat=no]).
instance(79, class=yes, [age=range40_49, menopause=premeno, tumorSize=range25_29,
                         invNodes=range0_2, nodeCaps=no, degMalig=3, breast=left,
                         breastQuad=right_up, irradiat=no]).
instance(80, class=no, [age=range60_69, menopause=ge40, tumorSize=range30_34,
                         invNodes=range6_8, nodeCaps=yes, degMalig=2, breast=right,
                         breastQuad=right_up, irradiat=no]).
instance(81, class=no, [age=range50_59, menopause=lt40, tumorSize=range15_19,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(82, class=no, [age=range40_49, menopause=premeno, tumorSize=range25_29,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=left_low, irradiat=no]).
instance(83, class=no, [age=range40_49, menopause=premeno, tumorSize=range30_34,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                         breastQuad=left_up, irradiat=no]).
instance(84, class=no, [age=range60_69, menopause=ge40, tumorSize=range15_19,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=left_up, irradiat=yes]).
instance(85, class=no, [age=range30_39, menopause=premeno, tumorSize=range0_4,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=central, irradiat=no]).
instance(86, class=no, [age=range50_59, menopause=ge40, tumorSize=range35_39,
                         invNodes=range0_2, nodeCaps=no, degMalig=3, breast=left,
                         breastQuad=left_up, irradiat=no]).
```

```
instance(87, class=no, [age=range40_49, menopause=premeno, tumorSize=range40_44,
                        invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                        breastQuad=left_up, irradiat=no]).
instance(88, class=no, [age=range30_39, menopause=premeno, tumorSize=range25_29,
                        invNodes=range6_8, nodeCaps=yes, degMalig=2, breast=right,
                        breastQuad=left_up, irradiat=yes]).
instance(89, class=no, [age=range50_59, menopause=ge40, tumorSize=range20_24,
                        invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                        breastQuad=left_low, irradiat=no]).
instance(90, class=no, [age=range50_59, menopause=ge40, tumorSize=range30_34,
                        invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                        breastQuad=left_up, irradiat=no]).
instance(91, class=yes, [age=range60_69, menopause=ge40, tumorSize=range20_24,
                        invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                        breastQuad=left_up, irradiat=no]).
instance(92, class=yes, [age=range30_39, menopause=premeno, tumorSize=range30_34,
                        invNodes=range3_5, nodeCaps=no, degMalig=3, breast=right,
                        breastQuad=left_up, irradiat=yes]).
instance(93, class=yes, [age=range50_59, menopause=lt40, tumorSize=range20_24,
                        invNodes=range0_2, nodeCaps=missing, degMalig=1, breast=left,
                        breastQuad=left_up, irradiat=no]).
instance(94, class=no, [age=range50_59, menopause=premeno, tumorSize=range10_14,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                        breastQuad=left_up, irradiat=no]).
instance(95, class=no, [age=range50_59, menopause=ge40, tumorSize=range20_24,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                        breastQuad=left_up, irradiat=no]).
instance(96, class=no, [age=range40_49, menopause=premeno, tumorSize=range45_49,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                        breastQuad=left_low, irradiat=yes]).
instance(97, class=yes, [age=range30_39, menopause=premeno, tumorSize=range40_44,
                        invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                        breastQuad=left_up, irradiat=no]).
instance(98, class=no, [age=range50_59, menopause=premeno, tumorSize=range10_14,
                        invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                        breastQuad=left_low, irradiat=no]).
instance(99, class=yes, [age=range60_69, menopause=ge40, tumorSize=range30_34,
                        invNodes=range0_2, nodeCaps=no, degMalig=3, breast=right,
                        breastQuad=left_up, irradiat=yes]).
instance(100, class=yes, [age=range40_49, menopause=premeno, tumorSize=range35_39,
                        invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                        breastQuad=left_up, irradiat=no]).
instance(101, class=yes, [age=range40_49, menopause=premeno, tumorSize=range20_24,
                        invNodes=range3_5, nodeCaps=yes, degMalig=2, breast=left,
                        breastQuad=left_low, irradiat=yes]).
instance(102, class=yes, [age=range50_59, menopause=premeno, tumorSize=range15_19,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                        breastQuad=left_low, irradiat=no]).
instance(103, class=no, [age=range50_59, menopause=ge40, tumorSize=range30_34,
                        invNodes=range0_2, nodeCaps=no, degMalig=3, breast=right,
                        breastQuad=left_low, irradiat=no]).
instance(104, class=no, [age=range60_69, menopause=ge40, tumorSize=range20_24,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                        breastQuad=left_up, irradiat=no]).
instance(105, class=no, [age=range40_49, menopause=premeno, tumorSize=range20_24,
                        invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                        breastQuad=right_low, irradiat=no]).
instance(106, class=yes, [age=range60_69, menopause=ge40, tumorSize=range30_34,
                        invNodes=range3_5, nodeCaps=yes, degMalig=2, breast=left,
                        breastQuad=central, irradiat=yes]).
instance(107, class=yes, [age=range60_69, menopause=ge40, tumorSize=range20_24,
                        invNodes=range3_5, nodeCaps=no, degMalig=2, breast=left,
                        breastQuad=left_low, irradiat=yes]).
instance(108, class=yes, [age=range50_59, menopause=premeno, tumorSize=range25_29,
                        invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                        breastQuad=right_up, irradiat=no]).
instance(109, class=no, [age=range50_59, menopause=ge40, tumorSize=range30_34,
                        invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                        breastQuad=right_up, irradiat=no]).
```

```
instance(110, class=no, [age=range40_49, menopause=premeno, tumorSize=range20_24,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=right_low, irradiat=no]).
instance(111, class=no, [age=range60_69, menopause=ge40, tumorSize=range15_19,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                         breastQuad=left_up, irradiat=no]).
instance(112, class=no, [age=range60_69, menopause=ge40, tumorSize=range30_34,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=left_low, irradiat=yes]).
instance(113, class=no, [age=range30_39, menopause=premeno, tumorSize=range30_34,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=left_up, irradiat=no]).
instance(114, class=no, [age=range30_39, menopause=premeno, tumorSize=range40_44,
                         invNodes=range3_5, nodeCaps=no, degMalig=3, breast=right,
                         breastQuad=right_up, irradiat=yes]).
instance(115, class=no, [age=range60_69, menopause=ge40, tumorSize=range5_9,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                         breastQuad=central, irradiat=no]).
instance(116, class=no, [age=range60_69, menopause=ge40, tumorSize=range10_14,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                         breastQuad=left_up, irradiat=no]).
instance(117, class=yes, [age=range40_49, menopause=premeno, tumorSize=range30_34,
                         invNodes=range6_8, nodeCaps=yes, degMalig=3, breast=right,
                         breastQuad=left_up, irradiat=no]).
instance(118, class=no, [age=range60_69, menopause=ge40, tumorSize=range10_14,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                         breastQuad=left_up, irradiat=no]).
instance(119, class=no, [age=range40_49, menopause=premeno, tumorSize=range35_39,
                         invNodes=range9_11, nodeCaps=yes, degMalig=2, breast=right,
                         breastQuad=left_up, irradiat=yes]).
instance(120, class=no, [age=range40_49, menopause=premeno, tumorSize=range20_24,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                         breastQuad=left_low, irradiat=no]).
instance(121, class=yes, [age=range40_49, menopause=premeno, tumorSize=range30_34,
                         invNodes=range0_2, nodeCaps=yes, degMalig=3, breast=right,
                         breastQuad=right_up, irradiat=no]).
instance(122, class=no, [age=range50_59, menopause=premeno, tumorSize=range25_29,
                         invNodes=range0_2, nodeCaps=yes, degMalig=2, breast=left,
                         breastQuad=left_up, irradiat=no]).
instance(123, class=no, [age=range40_49, menopause=premeno, tumorSize=range15_19,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(124, class=yes, [age=range30_39, menopause=premeno, tumorSize=range35_39,
                         invNodes=range9_11, nodeCaps=yes, degMalig=3, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(125, class=no, [age=range30_39, menopause=premeno, tumorSize=range10_14,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=right_low, irradiat=no]).
instance(126, class=no, [age=range50_59, menopause=ge40, tumorSize=range30_34,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                         breastQuad=left_low, irradiat=no]).
instance(127, class=no, [age=range60_69, menopause=ge40, tumorSize=range30_34,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=left_up, irradiat=no]).
instance(128, class=no, [age=range60_69, menopause=ge40, tumorSize=range25_29,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(129, class=yes, [age=range40_49, menopause=premeno, tumorSize=range15_19,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=left_up, irradiat=no]).
instance(130, class=no, [age=range60_69, menopause=ge40, tumorSize=range15_19,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=left_low, irradiat=no]).
instance(131, class=no, [age=range40_49, menopause=premeno, tumorSize=range30_34,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=right_low, irradiat=no]).
instance(132, class=no, [age=range20_29, menopause=premeno, tumorSize=range35_39,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=right_up, irradiat=no]).
```

```
instance(133, class=yes, [age=range40_49, menopause=premeno, tumorSize=range30_34,
                          invNodes=range0_2, nodeCaps=no, degMalig=3, breast=right,
                          breastQuad=right_up, irradiat=no]).
instance(134, class=yes, [age=range40_49, menopause=premeno, tumorSize=range25_29,
                          invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                          breastQuad=left_low, irradiat=no]).
instance(135, class=no, [age=range30_39, menopause=premeno, tumorSize=range30_34,
                          invNodes=range0_2, nodeCaps=no, degMalig=3, breast=left,
                          breastQuad=left_low, irradiat=no]).
instance(136, class=yes, [age=range30_39, menopause=premeno, tumorSize=range15_19,
                          invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                          breastQuad=left_low, irradiat=no]).
instance(137, class=no, [age=range50_59, menopause=ge40, tumorSize=range0_4,
                          invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                          breastQuad=central, irradiat=no]).
instance(138, class=no, [age=range50_59, menopause=ge40, tumorSize=range0_4,
                          invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                          breastQuad=left_low, irradiat=no]).
instance(139, class=yes, [age=range60_69, menopause=ge40, tumorSize=range50_54,
                          invNodes=range0_2, nodeCaps=no, degMalig=3, breast=right,
                          breastQuad=left_up, irradiat=no]).
instance(140, class=no, [age=range50_59, menopause=premeno, tumorSize=range30_34,
                          invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                          breastQuad=central, irradiat=no]).
instance(141, class=yes, [age=range60_69, menopause=ge40, tumorSize=range20_24,
                          invNodes=range24_26, nodeCaps=yes, degMalig=3, breast=left,
                          breastQuad=left_low, irradiat=yes]).
instance(142, class=no, [age=range40_49, menopause=premeno, tumorSize=range25_29,
                          invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                          breastQuad=left_up, irradiat=no]).
instance(143, class=yes, [age=range40_49, menopause=premeno, tumorSize=range30_34,
                          invNodes=range3_5, nodeCaps=no, degMalig=2, breast=right,
                          breastQuad=left_up, irradiat=no]).
instance(144, class=no, [age=range50_59, menopause=premeno, tumorSize=range20_24,
                          invNodes=range3_5, nodeCaps=yes, degMalig=2, breast=left,
                          breastQuad=left_low, irradiat=no]).
instance(145, class=no, [age=range50_59, menopause=ge40, tumorSize=range15_19,
                          invNodes=range0_2, nodeCaps=yes, degMalig=2, breast=left,
                          breastQuad=central, irradiat=yes]).
instance(146, class=no, [age=range50_59, menopause=premeno, tumorSize=range10_14,
                          invNodes=range0_2, nodeCaps=no, degMalig=3, breast=left,
                          breastQuad=left_low, irradiat=no]).
instance(147, class=yes, [age=range30_39, menopause=premeno, tumorSize=range30_34,
                          invNodes=range9_11, nodeCaps=no, degMalig=2, breast=right,
                          breastQuad=left_up, irradiat=yes]).
instance(148, class=no, [age=range60_69, menopause=ge40, tumorSize=range10_14,
                          invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                          breastQuad=left_low, irradiat=no]).
instance(149, class=no, [age=range40_49, menopause=premeno, tumorSize=range40_44,
                          invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                          breastQuad=left_low, irradiat=no]).
instance(150, class=no, [age=range50_59, menopause=ge40, tumorSize=range30_34,
                          invNodes=range9_11, nodeCaps=missing, degMalig=3, breast=left,
                          breastQuad=left_up, irradiat=yes]).
instance(151, class=yes, [age=range40_49, menopause=premeno, tumorSize=range50_54,
                          invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                          breastQuad=left_low, irradiat=yes]).
instance(152, class=no, [age=range50_59, menopause=ge40, tumorSize=range15_19,
                          invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                          breastQuad=right_up, irradiat=no]).
instance(153, class=no, [age=range50_59, menopause=ge40, tumorSize=range40_44,
                          invNodes=range3_5, nodeCaps=yes, degMalig=2, breast=left,
                          breastQuad=left_low, irradiat=no]).
instance(154, class=yes, [age=range30_39, menopause=premeno, tumorSize=range25_29,
                          invNodes=range3_5, nodeCaps=yes, degMalig=3, breast=left,
                          breastQuad=left_low, irradiat=yes]).
instance(155, class=no, [age=range60_69, menopause=ge40, tumorSize=range10_14,
                          invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                          breastQuad=left_low, irradiat=no]).
```

```
instance(156, class=no, [age=range60_69, menopause=lt40, tumorSize=range10_14,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                         breastQuad=right_up, irradiat=no]).
instance(157, class=yes, [age=range30_39, menopause=premeno, tumorSize=range30_34,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=left_up, irradiat=no]).
instance(158, class=yes, [age=range30_39, menopause=premeno, tumorSize=range20_24,
                         invNodes=range3_5, nodeCaps=yes, degMalig=2, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(159, class=no, [age=range50_59, menopause=ge40, tumorSize=range10_14,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                         breastQuad=left_up, irradiat=no]).
instance(160, class=no, [age=range60_69, menopause=ge40, tumorSize=range25_29,
                         invNodes=range0_2, nodeCaps=no, degMalig=3, breast=right,
                         breastQuad=left_up, irradiat=no]).
instance(161, class=no, [age=range50_59, menopause=ge40, tumorSize=range25_29,
                         invNodes=range3_5, nodeCaps=yes, degMalig=3, breast=right,
                         breastQuad=left_up, irradiat=no]).
instance(162, class=no, [age=range40_49, menopause=premeno, tumorSize=range30_34,
                         invNodes=range6_8, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=left_up, irradiat=no]).
instance(163, class=no, [age=range60_69, menopause=ge40, tumorSize=range50_54,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(164, class=no, [age=range50_59, menopause=premeno, tumorSize=range30_34,
                         invNodes=range0_2, nodeCaps=no, degMalig=3, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(165, class=yes, [age=range40_49, menopause=ge40, tumorSize=range20_24,
                         invNodes=range3_5, nodeCaps=no, degMalig=3, breast=right,
                         breastQuad=left_low, irradiat=yes]).
instance(166, class=yes, [age=range50_59, menopause=ge40, tumorSize=range30_34,
                         invNodes=range6_8, nodeCaps=yes, degMalig=2, breast=left,
                         breastQuad=right_low, irradiat=yes]).
instance(167, class=yes, [age=range60_69, menopause=ge40, tumorSize=range25_29,
                         invNodes=range3_5, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=right_up, irradiat=no]).
instance(168, class=no, [age=range40_49, menopause=premeno, tumorSize=range20_24,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=central, irradiat=no]).
instance(169, class=no, [age=range40_49, menopause=premeno, tumorSize=range20_24,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=left_up, irradiat=no]).
instance(170, class=no, [age=range40_49, menopause=premeno, tumorSize=range50_54,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(171, class=yes, [age=range50_59, menopause=ge40, tumorSize=range20_24,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=central, irradiat=no]).
instance(172, class=yes, [age=range50_59, menopause=ge40, tumorSize=range30_34,
                         invNodes=range3_5, nodeCaps=no, degMalig=3, breast=right,
                         breastQuad=left_up, irradiat=no]).
instance(173, class=no, [age=range40_49, menopause=ge40, tumorSize=range25_29,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(174, class=yes, [age=range50_59, menopause=premeno, tumorSize=range25_29,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                         breastQuad=left_up, irradiat=no]).
instance(175, class=no, [age=range40_49, menopause=premeno, tumorSize=range40_44,
                         invNodes=range3_5, nodeCaps=yes, degMalig=3, breast=right,
                         breastQuad=left_up, irradiat=yes]).
%
```

```
% **** Test Data
/*     =========

instance(176, class=no, [age=range40_49, menopause=premeno, tumorSize=range20_24,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=left_up, irradiat=no]).
instance(177, class=no, [age=range40_49, menopause=premeno, tumorSize=range20_24,
                         invNodes=range3_5, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=left_up, irradiat=no]).
instance(178, class=yes, [age=range40_49, menopause=premeno, tumorSize=range25_29,
                         invNodes=range9_11, nodeCaps=yes, degMalig=3, breast=right,
                         breastQuad=left_up, irradiat=no]).
instance(179, class=yes, [age=range40_49, menopause=premeno, tumorSize=range25_29,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=left_low, irradiat=no]).
instance(180, class=no, [age=range40_49, menopause=premeno, tumorSize=range20_24,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=right,
                         breastQuad=right_up, irradiat=no]).
instance(181, class=no, [age=range30_39, menopause=premeno, tumorSize=range40_44,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=right_up, irradiat=no]).
instance(182, class=yes, [age=range60_69, menopause=ge40, tumorSize=range10_14,
                         invNodes=range6_8, nodeCaps=yes, degMalig=3, breast=left,
                         breastQuad=left_up, irradiat=yes]).
instance(183, class=no, [age=range40_49, menopause=premeno, tumorSize=range35_39,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(184, class=yes, [age=range50_59, menopause=ge40, tumorSize=range30_34,
                         invNodes=range3_5, nodeCaps=no, degMalig=3, breast=left,
                         breastQuad=left_low, irradiat=no]).
instance(185, class=no, [age=range40_49, menopause=premeno, tumorSize=range5_9,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                         breastQuad=left_low, irradiat=yes]).
instance(186, class=no, [age=range60_69, menopause=ge40, tumorSize=range15_19,
                         invNodes=range0_2, nodeCaps=no, degMalig=1, breast=left,
                         breastQuad=right_low, irradiat=no]).
instance(187, class=no, [age=range40_49, menopause=premeno, tumorSize=range30_34,
                         invNodes=range0_2, nodeCaps=no, degMalig=3, breast=right,
                         breastQuad=right_up, irradiat=no]).
instance(188, class=yes, [age=range40_49, menopause=premeno, tumorSize=range25_29,
                         invNodes=range0_2, nodeCaps=no, degMalig=3, breast=left,
                         breastQuad=left_up, irradiat=no]).
instance(189, class=no, [age=range50_59, menopause=ge40, tumorSize=range5_9,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=right_up, irradiat=no]).
instance(190, class=no, [age=range50_59, menopause=premeno, tumorSize=range25_29,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=right,
                         breastQuad=right_low, irradiat=no]).
instance(191, class=yes, [age=range50_59, menopause=premeno, tumorSize=range25_29,
                         invNodes=range0_2, nodeCaps=no, degMalig=2, breast=left,
                         breastQuad=right_up, irradiat=no]).

% ---- End of Data File --------- */
```

ภาคผนวก ข

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

- N. Kerdprasop, S. Pilabutr, and K. Kerdprasop (2009). Improving medical database consistency with induced trigger rules. In K. Nakamatsu, G. Phillips-Wrens, L.C. Jain, R.J. Howlett (Eds.), *New Advances in Intelligent Decision Technologies*, pp. 265-274, Springer. (ISBN: 978-3-642-00908-2, DOI: 10.1007/978-3-642-00909-9).

- N. Kerdprasop, N. Muenrat, and K. Kerdprasop (2008). Decision rule induction in a learning content management system. *International Journal of Computer, Information, and Systems Science, and Engineering (IJCISSE)*, Volume 2, Number 2, pp. 77-81.

- N. Kerdprasop and K. Kerdprasop (2008). The design of an inductive database framework. *Proceedings of 1st Rajamangala University of Technology Conference*, Trung, Thailand, August 27-29.

- K. Kerdprasop, N. Kerdprasop and E. Pheddee (2005). A framework for inductive rule-based expert systems. *Proceedings of the NSTDA Annual Conference S&T in Thailand: Towards the Molecular Economy*, Science Park, Bangkok, Thailand, March 28-30, p.209.

Book chapter in K. Nakamatsu, G. Phillips-Wrens, L.C. Jain, R.J. Howlett (Eds.),
New Advances in Intelligent Decision Technologies, pp. 265-274, Springer.
(ISBN: 978-3-642-00908-2, DOI: 10.1007/978-3-642-00909-9)

Improving Medical Database Consistency with Induced Trigger Rules

Nittaya Kerdprasop, Sirikanjana Pilabutr and Kittisak Kerdprasop

Data Engineering and Knowledge Discovery (DEKD) Research Unit,
School of Computer Engineering, Suranaree University of Technology,
Nakhon Ratchasima 30000, THAILAND

Abstract The concept of triggers has been around for more than two decades. Despite their diverse potential usages, trigger rules are difficult to define correctly and have to be carefully hand-coded by database programmers. We suggest an automatic way of trigger rule creation by the advanced technology of data mining. We propose a framework of trigger rule induction as well as a method for trigger conflict resolution. On trigger firing the problem may arise if several trigger rules are eligible for execution. We propose a conflict resolution scheme that incorporates derived knowledge as a major part of the trigger rule prioritization. By means of trigger scheduling, deterministic behavior of the trigger processing can be guaranteed. We demonstrate the utilization of our proposed method on enhancing medical database consistency.

1 Introduction

A database is a collection of objects such as patient records together with a set of integrity constraints on these objects. Integrity constraints are predicates defined by the database designer as a requirement for database to be true on any database state. The values of objects in the database at any given time determine the state of the database. The state changes if there is a modification in the value of a database object. A database state is *consistent* if the values of the objects satisfy the specified integrity constraints. Database consistency is an important property to guarantee its reliability on any application.

To prevent the database from being inconsistent, data objects have to be accessed and modified only through the transactions. A transaction is a set of operations such as INSERT, DELETE, UPDATE that causes the database to change from one consistent state to another. On transaction processing, integrity constraints pertaining to the transaction are evaluated. If the constraints evaluate to false, called constraint violation, then the transaction that causes this event is undone. Integrity constraints are, however, capable of ensuring simple events such as domain integrity, referential integrity. To impose complex enforcement such as business rules or complicated update constraints across applications, trigger rules are deployed as a powerful and expressive tool to enforce integrity checking and thus enabling the filtering of state changes that violate database consistency.

Triggers, also known as event-condition-action (ECA) rules [26], are one major concept of active databases which extend traditional database systems with the mechanism to respond automatically to some specific events. The events may take place either inside or outside the database system. Upon the occurrence of the specified event, the rule condition is evaluated. If the condition is satisfied, then some actions are performed. Although triggers are regarded as an important database feature on consistency monitoring, their deployment is still limited. This is due to the fact that creating complex trigger rules is not an easy task [9, 17, 19]. Tools and environments to aid users and database programmers are certainly needed. It is thus our aim to provide a method to automatically generating trigger rules from current database contents by means of data mining techniques. The induced trigger rules can be viewed as supplementary constraints to help increasing database consistency.

This paper is organized as follows. After the introduction section, we review some background on triggers and their related issues. Section 3 is the proposed framework of inducing trigger rules from database contents and its application to medical database. We also provide an algorithm to solve trigger conflict problem together with detailed explanation of the algorithm in section 4. Section 5 discusses other work related to ours. Section 6 concludes the paper.

2 Triggers and related issues

In SQL standard [11, 14], triggers are expressed by means of event-condition-action rules, as presented in figure 1. Each trigger is identified by a name. It is possible to specify whether a trigger must be executed BEFORE or AFTER its triggering event. SQL triggers allow only the INSERT, DELETE, and UPDATE as triggering event, and limit to a single event be monitored per single trigger rule.

```
<trigger definition> ::= CREATE TRIGGER <trigger name>
                         { BEFORE | AFTER } <trigger event> ON <table>
                         [ REFERENCING <tran_table or var_list> ]
                         <triggered action>
<trigger event>   ::= INSERT | DELETE | UPDATE [ OF < column list> ]
<triggered action> ::= [ FOR EACH { ROW | STATEMENT } ]
                       [ WHEN ( <condition> ) ] <triggered SQL statement>
```

Fig. 1 Definition of SQL triggers.

The WHEN clause specifies an additional condition to be checked once the trigger rule is fired and before the action is executed. Conditions are predicates over the database state. If the WHEN clause is missing, the condition is supposed to be true and the trigger action is executed as soon as the trigger event occurs. The action is executed when the rule is triggered and its condition is true. Actions are stored procedures and may include SQL statements, control constructs, and calls to user-defined functions. The following example shows a trigger rule to impose a constraint on the database that the age of any person may never decrease.

Example 1: Trigger rule to guarantee no decrease on age value.

```
CREATE TRIGGER age-no-decrease BEFORE UPDATE OF Patient
FOR EACH ROW
  WHEN (new.Age < old.Age) begin log the event; signal error condition; end
```

The potential applications of triggers are significant [9, 19, 23]: signal integrity constraint violation and force rollbacks of the violating transactions, maintain consistency across system catalogs or other metadata, notify users in the form of messages, implement business rules or workflow management, and many more.

The behavior of triggers is defined as the "execution model." It specifies how trigger rules are evaluated and treated at runtime. Figure 2 illustrates the steps in processing triggers [22]. The signaling phase detects and signals the occurrence of an event. The event activates the corresponding trigger rules in the triggering phase, and the condition parts of the triggered rules are evaluated in the evaluation phase. The trigger conflict problem occurs when the conditions of more than one trigger rules are evaluated to be true. The scheduling phase indicates the order to process conflict triggers. The execution phase processes the scheduled trigger rules. On processing rule's action, the change in a database state may trigger another or even the same set of rules.

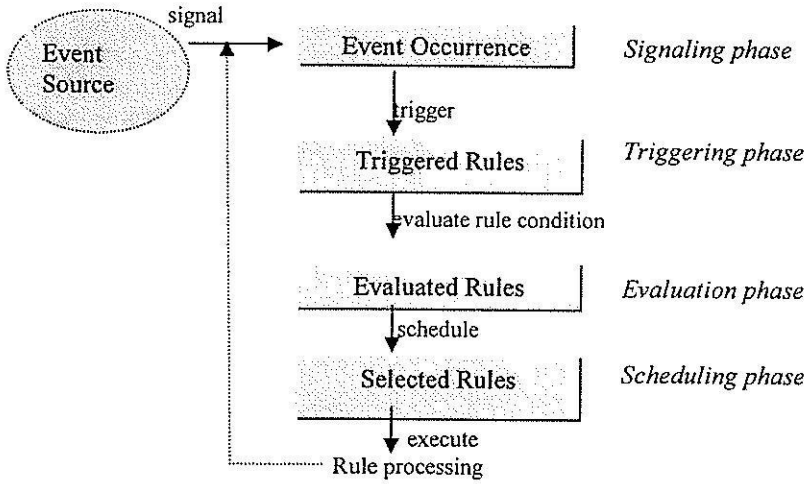


Fig. 2 Trigger rule execution steps.

After the evaluation phase, more than one rule may be eligible for execution. This problem is known as *trigger rule conflict* [4, 22]. To solve the problem, the database management system must provide a *conflict resolution policy* to select a trigger rule for execution. The common conflict resolution policy adopted by most systems is assigning rule priority [21, 26]. The rule prioritization is either assigning a high priority to the rule that is most recently fired, or setting priority on the specificity of the rule. The two approaches are dynamic, thus less practical in the system with large and complex trigger rule set. When deterministic behavior is highly desirable, the scheme to associate rules with priority statically is more appropriate. Static priority mechanism determines order of trigger rules either by the system (e.g., based on rule creation time) or by the user (e.g., explicitly associate each rule with a numeric value). We propose a conflict resolution mechanism (in section 4) to incorporate derived knowledge (i.e., the knowledge obtained from the database content) into the rule prioritization scheme.

3 The trigger induction framework and its application

We design the framework to add active behavior to the medical database through the induced trigger rules and rule processing module as shown in figure 3. There are three major components in our model: mining, trigger generation and conflict resolution components. Mining component induces knowledge in form of rules, association and classification, from the database contents. The data repository contains both base data and trigger rules. Trigger generation component is responsible for converting induced classification/association rules into trigger format then stores generated triggers in the repository. In case of trigger rule application and rule conflict occurs, conflict resolution component will handle the situation. We demonstrate the application of our proposed framework towards database consistency enforcement through example 2.

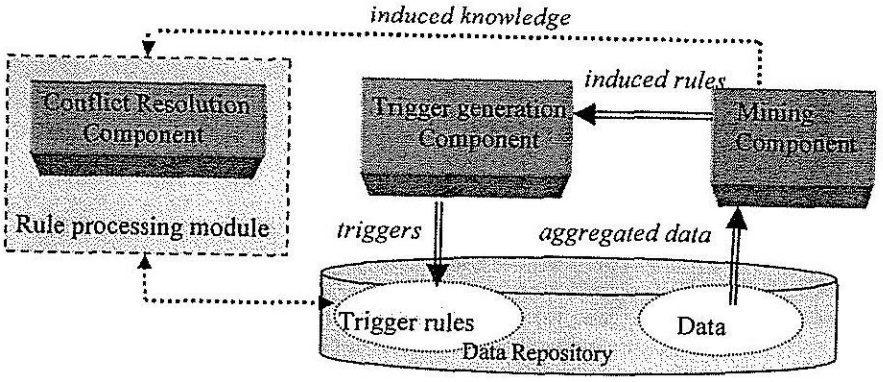


Fig. 3 A framework of active medical database containing trigger rule induction and trigger conflict resolution components.

Example 2: Trigger induction on Diabetes database.

For a given medical database containing base tables related to patient personal information and treatment records, the aggregated information collected from related base tables with selected features suitable for mining component is shown schematically as follows.

Diabetes (Patient_ID, Name, Sex, Age, Temperature, Blood_Pressure_Upper, Blood_Pressure_Low, Diabetes_family, Weight, Height, BMI, Blood_Sugar, Diabetes, Insulin_level)

These data are input to the mining component to induce the following classification rules:

1. If Diabetes_family=yes and BMI>24.9 Then Diabetes=yes
2. If Diabetes_family=no and blood_sugar>128 Then Diabetes=yes

The trigger generation component then converts these rules into SQL triggers.

```
CREATE TRIGGER rule_1 ON diabetes FOR UPDATE, INSERT
AS IF (SELECT COUNT(*) FROM diabetes
      WHERE (diabetes_family = 'yes') and (BMI > 24.9) and (diabetes <> 'yes') ) > 0
BEGIN ROLLBACK TRAN; RAISERROR ('diagnose error'); END


CREATE TRIGGER rule_2 ON diabetes FOR UPDATE, INSERT
AS IF (SELECT COUNT(*) FROM diabetes
WHERE (diabetes_family = 'no') and (blood_sugar > 128) and (diabetes <> 'yes')) > 0
BEGIN ROLLBACK TRAN; RAISERROR ('diagnose error'); END
```

Suppose there is an attempt to insert the following information into the database.

INSERT INTO Diabetics (Patient_ID, Name, Sex, Age, Temperature, Blood_Pressure_Upper, Blood_Pressure_Low, Diabetes_family, Weight, Height, BMI, Blood_Sugar, Diabetes, Insulin_level)

Values(P021, Amitta, Female, 33, 37.6, 130, 80, no, 72, 1.62, 27.4, 130, no, 0)

This new information violates Trigger rule_2 since *Diabetes_family = 'no'* and *Blood_Sugar = 130*, but the diagnosed *Diabetes = 'no'*. Therefore, this transaction has to be undone and the database remains in a consistent state.

The proposed framework is semi-automatic in that the trigger induction process has to be invoked by the database administrator. After database contents have been created and modified and numerous new contents might have been inserted into the database, the administrator may consider activating the trigger induction process. Upon activation the previous induced trigger rules are removed as they may not be relevant to the

current database state. The mining component has been set to induce only rules with 100% accuracy rate since precision is critical criteria in medical domain. Induced rules are prioritized in descending order of their support (or coverage) values. The *top-k* rules will be sent to trigger generation component; the k value is adjustable by the database administrator. At the moment we do not consider the issue of rule conflict since we assume that database integrity constraints are powerful mechanism sufficient to prevent conflicting cases inserted into database contents.

4 A method for trigger conflict resolution

After triggers have been created, the problem of trigger conflict might occur during trigger processing phase. In this section, we define an algorithm (in figure 4) to handle trigger conflict by reorganizing the trigger rules into different layers, or strata. Then, associate each stratum with a numeric priority. The major mechanism leading to priority assigning is the induced knowledge regarding the database state modification.

Input: an unordered stratum set S, a database R and its metadata
Output: an ordered stratum set O with assigned priority on each stratum
Steps:

1. $Active\_Tuple\_Set_i \leftarrow Activate(S_i, E)$
   /* Activate every stratum S_i, $S_i \in$ S, such that the occurrence of an event E can invoke its trigger rule(s), and record all affected tuples in the corresponding *Active_Tuple_Set*. */
2. $K_i \leftarrow Induce(Active\_Tuple\_Set_i)$
   /* The knowledge induction method (such as rule learning, decision-tree induction) is applied to induce knowledge from the content of each *Active_Tuple_Set*. The induced knowledge is stored in K_i. */
3. $q_i = Degree\_of\_Constraint\ (K_i, metadata)$
   /* Calculate the value q, or the *Degree_of_Constraint*, of each set of induced knowledge K comparing to the integrity constraints given as a metadata. */
4. *sort(i, q)*
   /* Apply any sorting algorithm on the q-value associated with each stratum S_i. */
5. *return an ordered stratum set* $O = \{ S_i \mid S$ *has been sorted by its index* $i \}$

Fig. 4 Trigger conflict resolution algorithm.

We have applied the concept of stratum [3] to guarantee that trigger rule execution eventually terminates. A *stratum* is an ordered set of trigger rules that locally converge. A stratum locally converges if after any transaction invoking trigger rules, rule processing terminates in a final state in which the set of triggered rules is empty. *Stratum set* is an unordered set of strata in which each stratum is independent from other strata so that the trigger rule execution in one stratum does not affect rules in other strata. The example of non-terminate trigger execution due to the cycle in action-triggering events is shown in figure 5. Breaking the cycle into different layers, depicted in figure 6, and the local convergence within each stratum are two sufficient conditions for termination on trigger execution.

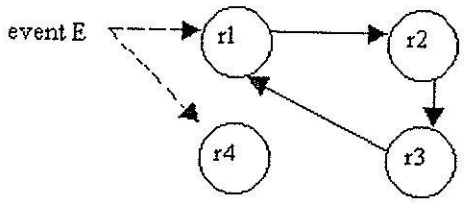


Fig. 5 The cycle among trigger rules.

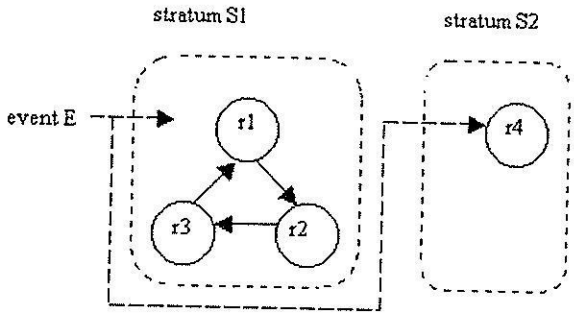


Fig. 6 Organizing cyclic rules into strata.

Step 3 of the proposed algorithm is to set priorities among strata when several strata are activated by the occurrence of an event. We propose the trigger conflict resolution algorithm to solve the problem of multiple stratum activation on an event E. The key concept is to apply, in step 2, knowledge induced from the database content and the metadata (i.e., the set of integrity constraints) to guide the priority assigning scheme. The function to compute priority for each stratum is defined as:

$$Degree\_of\_Constraint(K, IC) = \frac{\#matched\_rules}{\#total\_rules} + \alpha$$

where # *matched_rule* is the number of induced knowledge, represented in the format of rule, completely matching with the integrity constraint rule (i.e., NOT (k) AND c yields the contradiction when $k \in K$ and $c \in IC$),

total_rules is the total number of induced knowledge represented as rules,

α is the average accuracy of the induced knowledge rules normalized in order to prevent the domination of the accuracy over the proportion of *matched_rules* and *total_rules*.

Step 4 of the algorithm groups triggers into several strata, then each stratum will be sorted according to the *Degree_of_Constraint* values. At a final step, each stratum has been returned with the priority value associated with it.

5 Related work

The importance of integrating active behavior into the database systems has been recognized since the 1970s [12]. However, it was not until the late 1980s to early 1990s that the area of active databases has caught high interest among researchers [8, 10, 13, 22, 26]. At present most commercial database systems, such as Oracle, IBM DB2, Microsoft SQL Server, support the simple forms of triggers and also incorporate triggers into commercial object-oriented databases [7, 21]. The SQL standard [11, 14] has extensive coverage of triggers.

A simple concept of triggers has been successfully applied to solve problems in various domains [9]. For example, Sengupta et al [23] employ triggers to alert intrusion detection system administrator when a suspect

event has been detected. In medical domain the employment of triggers to achieve active behavior is quite rare. Most of the proposed methods are for detecting static events such as the discovery of relationships that suggest risks of adverse events in patient records [20, 24], detection of dependency patterns of process sequences for curing brain stroke patients [18], the generation of rules to annotated protein data in medical database [16], or the exploration of environmental health data [6]. Our work differ from those appeared in the literature in that we propose a framework of employing knowledge discovery techniques to automatically create trigger rules and also prioritize rules in case of conflict. The utilization of our proposed method is to increase consistency in medical database. Any database modification events violating constraints will be alerted and undone. The system designed by Agrawal and Johnson [1] is also to support medical database but in a different aspect; they concentrate on security and privacy preservation of patients and other sensitive health data.

Although trigger is a powerful mechanism in active database systems, designing and writing correct trigger rules are not an easy and straightforward task. The difficulty is due to the complex and sometime unpredictable behavior of the triggers. Poorly designed triggers can activate each other indefinitely, which leads to the non-terminate execution. Several methods have been proposed [2, 3, 4, 5, 15, 25] to analyze trigger behavior at compile time and runtime. There exist some work on developing tools to aid trigger designing semantically [19] and visually [17]. Another problem regarding trigger behavior is the deterministic property of the triggers. Deterministic trigger processing guarantees the same order of execution when several trigger rules are activated simultaneously. We propose a mechanism to utilize domain-knowledge in choosing among activated triggered rules.

6 Conclusions

We present the design framework of active medical databases. Active behavior of the database system has been obtained through a set of trigger rules. Triggers are both created manually by database programmers and induced automatically via data mining techniques. We devise a method to induce trigger rules from existing database contents. We also propose a method to solve trigger conflict problem.

The trigger rule conflict occurs when an event activates several trigger rules simultaneously. To maintain the deterministic property of the active database processing, the database management system has to provide a conflict resolution policy. The common policy adopted by most systems is to assign rule priority. The rule prioritization is based on either the recent update or the complexity of the rule's condition. We propose a different scheme of prioritization by taking into account the knowledge regarding the database state. Moreover, we consider priority at the level of stratum, which may contain several related triggers. The concept of stratification preserves the termination property of trigger rule processing.

The design of medical active database framework and the trigger-conflict-resolution algorithm is the preliminary work toward the design and implementation of a set of tools to help database designer on designing and analyzing a complex set of triggers. A further investigation on a more practical active database with a larger trigger set is necessary.

Acknowledgments The authors would like to thank all anonymous referees for their thorough reading and very helpful suggestion. This work has been fully supported by research fund from Suranaree University of Technology granted to the Data Engineering and Knowledge Discovery (DEKD) research unit. This research is also partly supported by grants from the National Research Council of Thailand (NRCT) and the Thailand Research Fund (TRF) under grant number RMU 5080026.

References

[1] Agrawal R, Johnson C (2007) Securing electronic health records without impeding the flow of information. Int. J Medical Informatics 76: 471-479

[2] Aiken A, Hellerstein JM, Widom J (1995) Static analysis techniques for predicting the behavior of active database rules. ACM Transactions on Database Systems 20(1): 3-41

[3] Baralis E, Ceri S, Paraboschi S (1996) Modularization techniques for active rule design. ACM Transactions on Database Systems 21(1): 1-29
[4] Baralis E, Ceri S, Paraboschi S (1998) Compile-time and runtime analysis of active behaviors. IEEE Transactions on Knowledge and Data Engineering 10(3): 353-370
[5] Baralis E, Widom J (1994) An algebraic approach to rule analysis in expert database system. In: Proc. 20th VLDB, pp.475-486
[6] Bedard Y, Gosselin P, Rivest S et al (2003) Integrating GIS components with knowledge discovery technology for environmental health decision support. Int. J Medical Informatics 70: 79-94
[7] Bertino E, Guerrini G, Merlo I (2000) Trigger inheritance and overriding in an active object database system. IEEE Transactions on Knowledge and Data Engineering 12(4): 588-608
[8] Buchmann A (1994) Current trends in active databases: Are we solving the right problems. In: Proc. Information Systems Design and Multimedia, pp.121-133
[9] Ceri S, Cochrane RJ, Widom J (2000) Practical applications of triggers and constraints: Successes and lingering issues. In: Proc. 26th VLDB, pp.254-262
[10] Chakravarthy S (1989) Rule management and evaluation: An active DBMS perspective. ACM SIGMOD Records 18(3): 20-28
[11] Eisenberg A, Melton J (2000) SQL:1999, formerly known as SQL3. ACM SIGMOD Records 28(1): 131-138
[12] Eswaran KP (1976) Specification, implementations and interactions of a trigger subsystem in an integrated database system. IBM Research Report RJ1820, San Jose, California
[13] Hanson EN, Widom J (1993) An overview of production rules in database systems. Knowledge Engineering Review 8(2): 121-143
[14] International Organization for Standardization, ISO/IEC 9075:2003
[15] Karadimce AP, Urban SD (1994) Conditional term rewriting as a formal basis for analysis of active database rules. In: Proc. Research Issues in Data Engineering (RIDE), pp.156-162
[16] Kretschmann E, Fleischmann W, Apweiler R (2001) Automatic rule generation for protein annotation with the C4.5 data mining algorithm applied on SWISS-PROT. Bioinformatics 17(10): 920-926
[17] Lee D, Mao W, Chiu H et al (2005) Designing triggers with trigger-by-example. Knowledge and Information Systems 7: 110-134
[18] Lin F, Chou S, Pan S et al (2001) Mining time dependency patterns in clinical pathways. Int. J Medical Informatics 62(1): 11-25
[19] Mota-Herranz L, Celma-Gimenez M (1998) Automatic generation of trigger rules for integrity enforcement in relational databases with view definition. In: Proc. 3rd Int. Conf. Flexible Query Answering Systems, pp.286-297
[20] Noren GN, Bate A, Hopstadius J et al (2008) Temporal pattern discovery for trends and transient effects: Its application to patient records. In: Proc. KDD, pp.963-971
[21] Paton N (1999) Active rules in database systems. Springer-Verlag
[22] Paton N, Diaz O (1999) Active database systems. ACM Computing Surveys 31(1): 63-103
[23] Sengupta S, Andriamanalimanana B, Card SW et al (2003) Towards data mining temporal patterns for anomaly intrusion detection systems. In: Proc. IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Tech. and Applications, pp.205-209
[24] Silva A, Cortez P, Santos MF et al (2008) Rating organ failure via adverse events using data mining in the intensive care unit. Artificial Intelligence in Medicine 43(3): 179-193
[25] van der Voort L, Siebes A (1993) Termination and confluence of rule execution. In: Proc. 2nd Int. Conf. Information and Knowledge Management, pp.245-255
[26] Widom J, Ceri S (1996) Active database systems: Triggers and rules for advanced database processing. Morgan Kaufmann

International Journal of Computer, Information, and Systems Science, and Engineering (IJCISSE),
Volume 2, Number 2, pp. 77-81

Decision Rule Induction in a Learning Content Management System

Nittaya Kerdprasop, Narin Muenrat, and Kittisak Kerdprasop

***Abstract*—**A learning content management system (LCMS) is an environment to support web-based learning content development. Primary function of the system is to manage the learning process as well as to generate content customized to meet a unique requirement of each learner. Among the available supporting tools offered by several vendors, we propose to enhance the LCMS functionality to individualize the presented content with the induction ability. Our induction technique is based on rough set theory. The induced rules are intended to be the supportive knowledge for guiding the content flow planning. They can also be used as decision rules to help content developers on managing content delivered to individual learner.

Keywords—Decision rules, Knowledge induction, Learning content management system, Rough set.

I. INTRODUCTION

THE term learning content management system (LCMS) refers to a suite of software tools designed to facilitate learning developers to create, manage and deliver learning content to distant learners [2]. The main features of an LCMS include content creation, content repository management, content delivery and interface, and learning process management such as course enrollment, assessment and performance tracking. An LCMS is adaptive and scalable in that creates proprietary content to meet the needs of individual learner.

The system offers course developers a feature to create and manage learning objects as customized content. Thus, the course development process can be viewed as a compilation of pieces of content retrieved from content repository to fit unique needs of different learners.

We, therefore, propose a knowledge induction technique to support course developers in designing flow of content appropriate to the ability of each learner. The induced knowledge is in the form of rules which are observed from the performance history of previous learners. These rules play the role of decision support in the course planning phase.

Decision support normally involves the integration of data and knowledge management to assist human on making effective and efficient choices [5], [16]. In the context of online course content delivery scalable to fit unique individual, decision making is on the basis of constant changing requirements that require a quick response. Traditional intuitive methods of decision making are no longer adequate to deal with such complicate situation. We consider rough set theory as a methodology to identify useful trends by exploring current and historical data.

Rough set theory is a new mathematical tool to deal with incomplete and inconsistent information. The theory was proposed by Pawlak, a Polish mathematician, in 1982 [12]. Since then it has drawn much attention from researchers interested in its theoretical aspects and applications [1], [4], [6], [8], [10], [18], [20]. Recent successful applications in the domains of machine learning and knowledge discovery have been reported [9], [11], [15], [19].

Manuscript received March 26, 2008. This work was supported in part by the National Research Council of Thailand. DEKD research unit is fully supported by Suranaree University of Technology.

Nittaya Kerdprasop is a principal researcher of DEKD research unit and an associate professor at the School of Computer Engineering, Suranaree University of Technology, 111 University Ave., Nakhon Ratchasima 30000, Thailand (e-mail: nittaya@sut.ac.th, nittaya.k@gmail.com).

Narin Muenrat is a master student of Computer Engineering school, Suranaree University of Technology.

Kittisak Kerdprasop is a director of the Data Engineering and Knowledge Discovery (DEKD) research unit, School of Computer Engineering, Suranaree University of Technology, 111 University Avenue, Muang District, Nakhon Ratchasima 30000, Thailand (phone: +66-44-224349; fax: +66-44-224602; e-mail: kerdpras@ sut.ac.th, KittisakThailand@gmail.com).

We study rough set theory within the framework of learning content management system. We focus on content creation that exploits rough set techniques as a tool to guide the decision on course content planning suitable to the learning performance of each learner.

The paper is organized as follows. Section 2 reviews the basic concepts of rough sets. Section 3 presents our idea of decision making with the induced patterns based on rough set approach. Section 4 illustrates the idea through the running examples. Section 5 concludes the paper.

II. PRELIMINARIES ON ROUGH SET

The notion of rough sets has been introduced by Zdzislaw Pawlak in the early 1980s [12], [13], [14] as a new concept of set with uncertain membership. Unlike fuzzy set, uncertainty in rough set theory does not need probability or the value of possibility to deal with vagueness. It is rather formalized through the simple concepts of lower and upper approximation, which are in turn defined on the basis of set. We present the basic concepts and terminology of rough sets within the framework of decision support system (DSS) [3].

Given the input data, the rough set-based DSS generates a list of certain and possible decision rules. The input data is a decision table comprising of conditional attributes, or conditions for short, and a decision attribute. Table 1 gives an example of a decision table containing information of eight students. Conditions are number of times the students log-in the system to access the online course and two pretest scores (intervals of numeric values). The level attribute (either basic or advanced) is a decision. Conditions together with decision attribute form a decision system.

TABLE I.

A DECISION TABLE JUDGING STUDENTS' PERFORMANCE

	Conditions			*Decision*
	log-in (c1)	score1 (c2)	score2 (c3)	level
s1	15	0-20	0-20	Basic
s2	15	0-20	21-40	Basic
s3	20	0-20	41-60	Basic
s4	20	0-20	41-60	Basic
s5	15	0-20	81-100	Advanced
s6	15	41-60	41-60	Advanced
s7	15	21-40	61-80	Advanced
s8	20	21-40	21-40	Advanced

A decision table is a representation of real-world data. Each row represents one object. Rough set theory is based on the formation of equivalence relations [7], [17] within the given data.

Definition 1: A *decision system* is any system of the form $A = \langle U, A, d \rangle$, where U is a non-empty finite set of objects called the universe, A is a non-empty finite set of conditions, and $d \notin A$ is the decision attribute.

Definition 2: Given a decision system $A = \langle U, A, d \rangle$, then with any $B \subseteq A$ there exists an *equivalence* or *indiscernibility relation* $I_A(B)$ such that

$$I_A(B) = \{(x, x') \in UxU \mid \forall a \in B\ [a(x) = a(x')]\ \}.$$

From the data samples in Table1, the followings are equivalent relations.

I(c1) = {{s1, s2, s5, s6, s7}, {s3, s4, s8}}
I(c2) = {{ s1, s2, s3, s4, s5}, {s6}, {s7, s8}}
I(c3) = {{s1},{s2, s8},{s3, s4, s6}, {s5}, {s7}}
I(c1,c2) = {{s1,s2,s5},{s3,s4},{s6},{s7},{s8}}
I(c1,c3) = {{s1},{s2},{s3,s4},{s5},{s6},{s7},{s8}}
I(c2,c3) = {{s1},{s2},{s3,s4},{s5},{s6},{s7},{s8}}
I(c1,c2,c3) = {{s1},{s2},{s3,s4},{s5},{s6},{s7},{s8}}

These equivalence relations partition the universe into groups of similar objects based on the values of some attributes. The question often arises is whether one can remove some attributes and still preserve the same equivalence relations. This question leads to the notion of *reduct* [7].

Definition 3: Let A = <U, A, d> be a decision system and P, Q ⊆ A be sets of conditions, P ≠ Q. The set P is the *reduct* of set Q if P is minimal (i.e. no redundant attributes in P) and the equivalence relations defined by P and Q are the same.

It can be seen from the listed equivalence relations that I(c1,c3) = I(c2,c3) = I(c1,c2,c3). Therefore, (log-in, score1) and (score1, score2) are reducts of (log-in, score1, score2). Either reduct can be used as a representative set of attributes. The intersection of all reducts produces *core attributes*. According to our example, score2 is a core attribute. A reduct table of (score1, score2) and its partitions are shown in Figure1.

	score	score2	level
s	0-20	0-20	basic
1	0-20	21-40	basic
2	0-20	41-60	basic
3	0-20	41-60	basic
4	0-20	81-100	advanced
5	41-60	41-60	advanced
6	21-40	61-80	advanced
7	21-40	21-40	advanced
8			

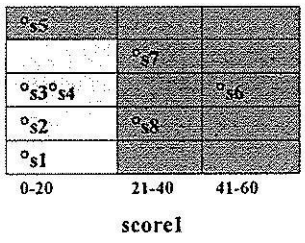


Figure 1. A reduct table and its partition into equivalence relations, each equivalence relation is represented by a rectangular region.

If we are interested in the decision criteria for the advanced-level students, we can infer decision rules from Figure1 as follows.

IF (score1=0-20 ∧ score2=81-100) THEN level = advanced

IF (score1=21-40 ∧ score2=21-40) THEN level = advanced

IF (score1=21-40 ∧ score2=61-80) THEN level = advanced

IF (score1=41-60 ∧ score2=41-60) THEN level = advanced

The decision criteria for basic-level students can be inferred accordingly.

IF (score1=0-20 ∧ score2=0-20) THEN level = basic

IF (score1=0-20 ∧ score2=21-40) THEN level = basic

IF (score1=0-20 ∧ score2=41-60) THEN level = basic

Suppose we are given additional information of the ninth student as shown in Figure2, then the above decision rules for the advanced-level students is no longer valid. It can be seen from Figure2 that s8 and s9 are in the same equivalence relation but their performance levels are different. It is such conflicting cases that

inspire the rough set concept. Given the two decision sets of advanced/basic level, the uncertain cases such as s8 and s9 can be approximated their membership by means of lower and upper approximation [7].

Definition 4: Let A = <U, A, d> be a decision system, $B \subseteq A$, $X \subseteq U$ and $[x]_B$ denote the equivalence class of $I_A(B)$. The *B-lower approximation* and *B-upper approximation* of X, denoted by bX and BX respectively, are defined by $bX = \{x \mid [x]_B \subseteq X\}$ and $BX = \{x \mid [x]_B \cap X \neq \varnothing\}$.

	score1	score2	level
s1	0-20	0-20	basic
s2	0-20	21-40	basic
s3	0-20	41-60	basic
s4	0-20	41-60	basic
s5	0-20	81-100	advanced
s6	41-60	41-60	advanced
s7	21-40	61-80	advanced
s8	21-40	21-40	advanced
s9	21-40	21-40	basic

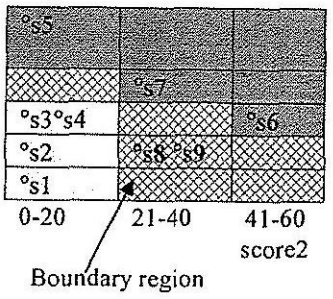


Figure 2. A decision table with conflict cases on students' performance level.

Given the information as shown in Figure2, B = {score1, score2} and X = {s5, s6, s7, s8} is set of students with advanced-level performance, then the lower approximation bX = {s5, s6, s7} and the upper approximation BX = {s5, s6, s7, s8, s9}. That is, the lower approximation of X is the set of all objects that are certainly belong to X. This set is also called *B-positive region* of X. The *B-negative region* of X is defined as U–BX, i.e. {s1, s2, s3, s4} or the set of all objects that definitely not belong to X. The area between these two sets is called *B-boundary region* of X, denoted by *BN*, and defined as $BN = BX–bX$. It is the set of all objects that cannot be classified as not belonging to X.

If the boundary region is empty, it is a *crisp* (precise) set; otherwise, the set is *rough*. The set of advanced-level students in Figure1 is a crisp set, but it is a rough set in Figure2. Decision rules generated from a rough set comprise of certain rules generated from the positive and negative regions, and possible rules generated from the boundary region.

A method to generate decision rules as explained above is static. The decision attribute is defined in advance. Within the framework of DSS that decision problems are usually not known in advance, such classical rough set methodology is inadequate. We thus propose in the next section our method of dynamic decision rule induction.

III. ROUGH SET BASED RULE INDUCTION

In the traditional environment of DSS, decision rules are manually encoding by knowledge engineers. It is a time consuming process that requires close collaboration between experts of the field and the computer professionals. With the emerging of data warehousing technology, we have a huge valuable resource of knowledge from which we can induce useful decision rules. But with the classical rough set method, the number of generated decision rules is tremendous. We propose a different approach of inducing certain and possible decision rules; the induction process is triggered by the query. The information regarding learners is stored in a table form and decision rules on content management guiding are induced by posing query on any students' attribute. By this scheme, we can limit the induction to only relevance rules. The framework of our approach is shown in Figure3.

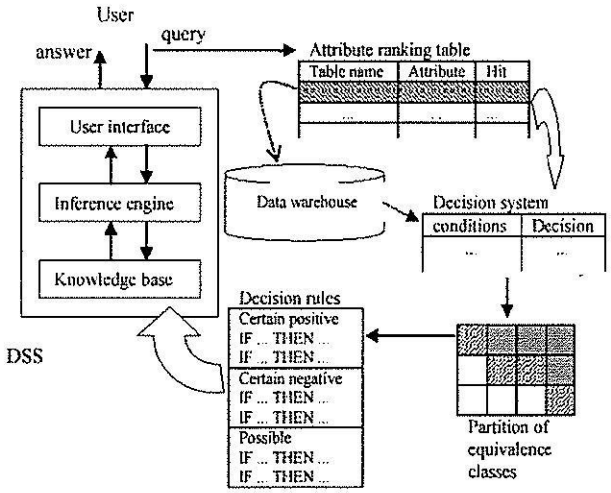


Figure 3. The induction of rough and precise knowledge.

Our proposed framework of decision rule induction is invoked by query. Once the query has been posted, the auxiliary data structure named *attribute ranking table* has been updated with the table's and attribute's name extracted from the query. The column *hit* counts the number of times that attributes has been used. The counter is sorted in descending order to place the most frequently used attribute in the first row. This attribute always referred to by users' queries, therefore it is worth generating decision rules based on this attribute value. The approach of inducing decision rules based on the most frequently asked attribute is described in the following algorithm.

Algorithm Decision rule induction

Input: User's query and a data warehouse
Output: Decision rules

Step:

1. Extract table names T_i and attribute names A_j from the query
2. Access the attribute ranking (AR) table and update the hit counter identified by each T_i and A_j
3. Descending sort the AR-table according to the hit value
4. Extract the top row of AR-table to obtain T_1 and A_1
5. Create a decision table A = <U, A, d> where d = A_1, A = a set of attributes in T_1, U = a set of records in T_1
6. Pre-process A by
   - removing attributes with number of distinct values = $|T_1|$
   - discretizing attributes with real values
7. Partition U into equivalence classes
8. Search for the first reduct R
9. Identify *bX*, *BX*, *BN* regions
10. From R, *bX*, *BX*, and *BN*, generate certain, negative, possible rules
11. Generalize all three classes of decision rules using dimension tables and hierarchical information from the data warehouse
12. Insert rules into the knowledge base

IV. RUNNING EXAMPLES

We use the student data shown in Table1 with additional record <s9, 20, 21-40, 21-40, basic> as our running example. The hierarchical information on interval order that 81-100 > 61-80 > 41-60 > 21-40 > 0-20 is used as background knowledge for decision rule generalization.

Example 1: Suppose there is a query consulting the system whether the score1 = 55 is high enough for the assigning the student to the advanced level.

Method:

(1) This query asks about student's level with score1 as a condition. Hence, a reduct decision table as in Figure2 is constructed.

(2) Then, the following decision rules are generated.

Certain positive rules

IF (score1=0-20 ∧ score2=81-100) THEN level = advanced

IF (score1=21-40 ∧ score2=61-80) THEN level = advanced

IF (score1=41-60 ∧ score2=41-60) THEN level = advanced

Certain negative rules

IF (score1=0-20 ∧ score2=0-20) THEN level = basic

IF (score1=0-20 ∧ score2=21-40) THEN level = basic

IF (score1=0-20 ∧ score2=41-60) THEN level = basic

Possible rules

IF (score1=21-40 ∧ score2=21-40) THEN level = advanced

(3) The three classes of decision rules are generalized according to the background knowledge. The final decision rules are as follow.

R1: IF (score1 > 20 ∧ score2 > 60) THEN level = advanced

R2: IF (score1 > 40 ∧ score2 > 40) THEN level = advanced

R3: IF (score1 > 20 ∧ score2 > 20) THEN level = possibly advanced

Notice that with the given information there is no matching rules from the negative class and R2 can be applied to answer this query.

Answer:

IF score2 > 40 THEN level = advanced.

IF score2 > 20 THEN level = possibly advanced.

Example 2: From the response of example 1, suppose the user wants to consult the system further that based on the information of her first pretest score, could the system predicts her second pretest score.

Method:

(1) The query asks the value of score2, given the value of score1=55. Thus, a decision attribute is score2 and a decision table is as shown in Table2.

TABLE II.

A DECISION TABLE WITH RESPECT TO EXAMPLE 2

	Conditions			*Decisio*
	log-in	score1	level	score2
s1	15	0-20	Basic	0-20
s2	15	0-20	Basic	21-40
s3	20	0-20	Basic	41-60
s4	20	0-20	Basic	41-60
s5	15	0-20	Advanced	81-100
s6	15	41-60	Advanced	41-60
s7	15	21-40	Advanced	61-80
s8	20	21-40	Basic	21-40
s9	20	21-40	Advanced	21-40

(2) There is no reduct. So, all conditional attributes are used in the approximation of *bX*, *BX*, and *BN* regions. The decision objectives (X) are set of all students whose score2 values are in the range 0-20, 21-40, 41-60, 61-80, and 81-100. From the approximation, these rules are induced:

Certain rules

IF (log-in=15 ∧ level = advanced ∧ score1=0-20) THEN score2= 81-100

IF (log-in=15 ∧ level = advanced ∧ score1=21-20) THEN score2=61-80

IF (log-in=20 ∧ level = basic ∧ score1=0-20) THEN score2=41-60

IF (log-in=15 ∧ level = advanced ∧ score1=0-20) THEN score2=41-60

IF (log-in=20 ∧ score1=21-40) THEN score2=21-40

Possible rules

IF (log-in=19 ∧ score1=20 ∧ level=basic) THEN score2=0-20 ∨ 21-40

(3) Generalized decision rules are as follow.

R1: IF (score1 = 0-20) THEN score2 = 81-100

R2: IF (score1 = 21-40) THEN score2 = 61-80

R3: IF (score1 = 0-20 ∨ 41-60) THEN score2 = 41-60

R4: IF (log-in=20 ∧ score1 = 21-40) THEN score2 = 21-40

R5: IF (log-in=15 ∧ score1 = 20 ∧ level=basic) THEN possibly score2 = 0-40

Answer:

IF score1 = 55 THEN score2 = 41-60.

V. CONCLUSIONS

In this paper we propose a technique of decision rule induction to induce knowledge that can facilitate the content management in the learning content management system. The induction process is based on the rough set theory. Our assumption is that system with the availability of a warehouse as a data and knowledge repository may produce tremendous amount of decision rules. We thus limit the number of rules by inducing only rules that are relevant to user's need. Relevancy is guided by query predicates. We propose the framework of the system and the algorithm of decision rule induction. The intuitive idea is illustrated trough running examples. Our proposed idea is general, so it can be applied to any kind of domain. We plan to test the effectiveness of our framework with the real-world data in the future.

REFERENCES

[1] Y. Caballero, R. Bello, A. Taboada, A. Nowe, M. Garcia, and G. Casas, "A new measure based in the rough set theory to estimate the training set quality", *Proc.8th Int. Symp. Symbolic and Numeric Algorithms for Scientific Computing*, pp.133-140, 2006.

[2] B. Chapman and B. Hall, *Learning Content Management System.* Brandonhall.com, New York,

[3] Z0Chen, *Computational Intelligence for Decision Support*, CRC Press, 2000.

[4] K. Cios, W. Pedrycz, and R. Swiniarski, *Data Mining Methods for Knowledge Discovery*, Kluwer Academic Publishers, 1998.

[5] C.W. Holsapple and A.B. Whinston, *Decision Support Systems: A Knowledge-Based Approach*, West Publishing Company, 1996.

[6] X. Hu, "Using rough sets theory and database operations to construct a good ensemble of classifiers for data mining application", *Proc.IEEE ICDM*, pp.233-240, 2001.

[7] J. Komorowski, L. Polkowski, and A. Skowron, "Rough sets: A tutorial", In S.K. Pal and A. Skowron (eds.), *Rough Fuzzy Hybridization: A New Trend in Decision-Making*, Springer, pp.3-98, 1999.

[8] A. Lenarcik and Z. Piasta, "Probabilistic rough classifiers with mixture of discrete and continuous variables", In T.Y. Lin and N. Cercone (eds.), *Rough Sets and Data Mining: Analysis for Imprecise Data*, Kluwer Academic Publishers, pp.373-383, 1997.

[9] D. Miao and L. Hou, "A comparison of rough set methods and representative inductive learning algorithms", *Fundamenta Informaticae*, vol.59, pp.203-218, 2004.

[10] P. Pattaraintakorn, N. Cercone, and K. Naruedomkul, "Hybrid intelligent systems: Selecting attributes for soft computing analysis", *Proc.29th Int. Conf. Computer Software and Applications*, pp.319-325, 2006.

[11] S. Pal and P. Mitra, "Case generation using rough sets with fuzzy representation", *IEEE Trans. Knowledge and Data Engineering*, vol.16, no.3, pp.292-300, 2004.

[12] Z. Pawlak, "Rough sets", *Int. Journal of Information and Computer Science*, vol.11, no.5, pp.341-356, 1982.

[13] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, 1991.

[14] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, and W. Ziarko, "Rough sets", *Communications of the ACM*, vol.38, no.11, pp.88-95, 1995.

[15] J. Peters, D. Lockery, and S. Ramanna, "Monte Carlo off-policy reinforcement learning; A rough set approach", *Proc. 5th Int. Conf. Hybrid Intelligent Systems*, pp.187-192, 2005.

[16] F. Radermacher, "Decision support systems: Scope and potential", *Decision Support Systems*, vol.12, pp.257-265, 1994.

[17] A. Skowron and C. Rauszer, "The discernibility matrices and functions in information systems", In R. Slowinski (ed.), *Intelligent Decision Support, Handbook of Applications and advances of the Rough Set Theory*, Kluwer Academic Publishers, pp.331-362, 1992.

[18] R. Swiniarski, "Rough sets and principal component analysis and their applications in feature extraction and selection, data model building and classification", In S. Pal and A. Skowron (eds.), *Fuzzy Sets, Rough Sets and Decision Making Processes*, Springer, 1998.

[19] L. Yang and L. Yang, "Study of a cluster algorithm based on rough sets theory", *Proc. 6th Int. Conf. Intelligent Systems Design and Applications*, pp.492-496, 2006.

[20] W. Ziarko, "The discovery, analysis, and representation of data dependencies in databases", In G. Piatetsky-Shapiro and W.J. Frawley (eds.), *Knowledge Discovery in Databases*, AAAI Press, pp.195-209, 1991.

Nittaya Kerdprasop is an associate professor at the school of computer engineering, Suranaree University of Technology, Thailand. She received her B.S. from Mahidol University, Thailand, in 1985, M.S. in computer science from the Prince of Songkla University, Thailand, in 1991 and Ph.D. in computer science from Nova Southeastern University, USA, in 1999. She is a member of ACM and IEEE Computer Society. Her research of interest includes Knowledge Discovery in Databases, AI, Logic Programming, Deductive and Active Databases.

Narin Muenrat received his bachelor degree in Information Technology in 2003 from Suranaree University of Technology. He is currently pursuing a master degree in computer engineering at Suranaree University of Technology. His research interests are content and learning management systems, web technology and open source programming.

Kittisak Kerdprasop is an associate professor at the school of computer engineering, Suranaree University of Technology, Thailand. He received his bachelor degree in Mathematics from Srinakarinwirot University, Thailand, in 1986, master degree in computer science from the Prince of Songkla University, Thailand, in 1991 and doctoral degree in computer science from Nova Southeastern University, USA., in 1999. His current research includes Data mining, Artificial Intelligence, Functional Programming, Computational Statistics.

Proceedings of 1st Rajamangala University of Technology Conference,
Trung, Thailand, August 27-29

การออกแบบกรอบแนวคิดของฐานข้อมูลเชิงอุปนัย
The Design of an Inductive Database Framework

นิตยา เกิดประสพ และ กิตติศักดิ์ เกิดประสพ
Nittaya Kerdprasop and Kittisak Kerdprasop

บทคัดย่อ

ฐานข้อมูลเชิงอุปนัยแตกต่างจากฐานข้อมูลเชิงสัมพันธ์ที่ใช้อยู่โดยทั่วไป ตรงที่ฐานข้อมูลเชิงอุปนัยบันทึกทั้งข้อมูลและแพทเทิร์น(หรือรูปแบบ)ข้อมูล ในระยะแรกของการนำเสนอแนวคิดเกี่ยวกับฐานข้อมูลเชิงอุปนัยมีวัตถุประสงค์หลักเพียงเพื่อให้ฐานข้อมูลสามารถสนับสนุนงานด้านการทำเหมืองข้อมูล แต่ผู้วิจัยมีแนวคิดที่จะออกแบบให้ทั้งระบบฐานข้อมูลและระบบการทำเหมืองข้อมูลสนับสนุนซึ่งกันและกัน โดยในการออกแบบมุ่งเน้นให้สามารถค้นหารูปแบบจากข้อมูล และสามารถนำทั้งข้อมูลและรูปแบบข้อมูลมาใช้ช่วยในการปรับปรุงรูปแบบข้อคำถามและตอบข้อคำถามของผู้ใช้ โดยผลการทดลองเบื้องต้นยืนยันว่าแนวความคิดนี้เป็นประโยชน์ในการเพิ่มประสิทธิภาพการตอบข้อคำถามของระบบฐานข้อมูล

คำสำคัญ : ฐานข้อมูลเชิงอุปนัย, การทำเหมืองข้อมูล

ABSTRACT

Inductive databases can be viewed as a natural extension of traditional databases to contain not only persistent data but also the generalization of stored data, which are called patterns. The idea of inductive databases has been proposed originally as a support system for the knowledge discovery or data mining process. We perceive the concept of inductive databases in a different angle. Instead of designing yet another inductive database system, we are looking for the deployment of an existing inductive query language and environment to support the database tasks. We focus on the task of query answering which has a high potential of being a beneficiary of the stored patterns in inductive databases. Our experimental results of query rewriting technique using induced patterns as a semantic knowledge confirm this advantage.

Key words : Inductive databases, Data mining

หน่วยวิจัยวิศวกรรมข้อมูลและการค้นหาความรู้ สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
Data Engineering and Knowledge Discovery Research Unit, School of Computer Engineering, Suranaree Univ. of Tech.
Corresponding author. E-mail: nittaya@sut.ac.th

บทนำ

การทำเหมืองข้อมูล (data mining) หรือการค้นหาความรู้จากฐานข้อมูล (knowledge discovery in databases) เป็นงานวิจัยในสาขาใหม่ที่ได้รับความสนใจอย่างมากจากนักคอมพิวเตอร์ในช่วงทศวรรษที่ผ่านมา ความนิยมนี้มีสาเหตุหลักมาจากการบันทึกข้อมูลลงในระบบฐานข้อมูลกระทำกันอย่างแพร่หลาย ทำให้เกิดข้อมูลอิเล็กทรอนิกส์เป็นปริมาณมหาศาล แต่การใช้ประโยชน์จากข้อมูลเหล่านี้ยังมีน้อยมาก เนื่องจากข้อจำกัดด้านกำลังคนที่จะทำหน้าที่วิเคราะห์ข้อมูลให้ได้ผลผลิตเป็นความรู้ใหม่ที่จะใช้ประโยชน์ได้ต่อไป ดังนั้นเมื่อมีการพัฒนาเทคนิคการทำเหมืองข้อมูลให้สามารถค้นหารูปแบบข้อมูล (patterns or models) ได้โดยอัตโนมัติ จึงทำให้เกิดความคาดหวังว่างานวิเคราะห์เพื่อหารูปแบบข้อมูล จะทำได้รวดเร็วขึ้นและลดภาระของนักวิเคราะห์ข้อมูลให้น้อยลง ผลที่ตามมาคือจะสามารถนำรูปแบบข้อมูลที่ค้นพบไปใช้ประโยชน์ด้านต่าง ๆ ได้รวดเร็วทันต่อความต้องการ

ตัวอย่างความสำเร็จของการใช้ประโยชน์จากเทคโนโลยีการทำเหมืองข้อมูล ได้แก่ การค้นพบพฤติกรรมผู้บริโภคจากฐานข้อมูลการชำระเงิน ณ จุดขาย (point-of-sale) ของลูกค้าในห้างสรรพสินค้าวอลมาร์ต ประเทศสหรัฐอเมริกา การค้นพบรูปแบบพฤติกรรมนี้นำไปสู่การวางแผนที่ดีขึ้นทั้งในด้านการจัดวางสินค้า การจัดการสินค้าคงคลัง รวมไปถึงการวางแผนขนส่งเพื่อกระจายสินค้าไปยังสาขาต่าง ๆ ทำให้วอลมาร์ตสามารถบริหารต้นทุนสินค้าได้อย่างมีประสิทธิภาพ

ถึงแม้จะปรากฏรายงานจำนวนมากที่ชี้ให้เห็นถึงความสำเร็จ และประโยชน์ที่ได้จากการทำเหมืองข้อมูลกับข้อมูลหลากหลายประเภท เช่น ข้อมูลธุรกิจ ข้อมูลทางวิทยาศาสตร์ ข้อมูลทางการแพทย์โดยเฉพาะข้อมูลจีโนมหรือรหัสพันธุกรรม แต่กระบวนการทำเหมืองข้อมูลยังเป็นเทคโนโลยีที่ต้องอาศัยผู้ชำนาญทางด้านนี้โดยเฉพาะ ในปัจจุบันยังไม่สามารถพัฒนาระบบให้ใช้งานได้ง่ายสำหรับผู้ใช้ทั่วไป เนื่องจากการทำเหมืองข้อมูลเป็นกระบวนการทำซ้ำ ที่ต้องมีการปรับปรุงทั้งโครงสร้างและเนื้อหาข้อมูลในแต่ละรอบของการทำงานเพื่อการค้นหารูปแบบที่มีความถูกต้องสูง และเป็นรูปแบบที่น่าสนใจนำไปใช้ให้เกิดประโยชน์ได้จริง

ความก้าวหน้าของการทำเหมืองข้อมูลในปัจจุบัน ถึงแม้ว่ายังไม่สามารถพัฒนากระบวนการไปสู่ลักษณะสำเร็จรูปที่ใช้งานได้ทันทีในแบบ plug-and-play ได้ แต่ก็ได้มีความพยายามที่จะรวบรวมขั้นตอนต่างๆ ของการทำเหมืองข้อมูลที่ในปัจจุบันเป็นขั้นตอนที่แยกจากกัน แต่ละขั้นตอนเป็นอิสระสามารถใช้วิธีการจัดการที่ต่างกันได้ ให้มาอยู่ในระบบปิดที่มีพื้นฐานแนวคิดเดียวกันมีวิธีการจัดการกับข้อมูลและรูปแบบข้อมูลที่เป็นมาตรฐานเดียวกัน โดยในปี ค.ศ.1996 T. Imielinski และ H. Mannila [22] ได้เสนอแนวคิดให้มีการเพิ่มฟังก์ชันการเรียนรู้รูปแบบข้อมูล (pattern mining) ลงในระบบจัดการฐานข้อมูล (database management system – DBMS) และเพิ่มชุดคำสั่ง SQL ให้โปรแกรมเมอร์สามารถระบุเทคนิคในการทำ mining สามารถบันทึกรูปแบบข้อมูลที่ค้นพบเก็บไว้ในฐานข้อมูล และสามารถใช้ภาษา SQL สอบถามรูปแบบข้อมูลที่ต้องการได้

T. Imielinski และ H. Mannila เสนอว่าการรวมความสามารถด้าน pattern mining เข้ากับ DBMS นี้ควรจะเป็นลักษณะ tightly coupling แทนการเพิ่มโมดูลในการทำ mining เป็นระดับชั้นเหนือ DBMS ในลักษณะ loosely coupling การขยายขีดความสามารถของระบบจัดการฐานข้อมูลให้สามารถจัดเก็บและทำงานได้กับทั้งข้อมูล (data) และรูปแบบข้อมูล (patterns) ทำให้เกิดเป็นแนวคิดของฐานข้อมูลชนิดใหม่ เรียกว่า **ฐานข้อมูลเชิงอุปนัย** (inductive databases)

ในช่วงต้นทศวรรษที่ 2000 จนกระทั่งถึงปัจจุบันนักพัฒนาโปรแกรมได้มีความพยายามจะปรับปรุงระบบจัดการฐานข้อมูลเชิงสัมพันธ์ที่นิยมใช้อยู่ในปัจจุบัน เช่น IBM DB2, Microsoft SQL Server และ Oracle ให้มีฟังก์ชันและชุดคำสั่งที่ทำงานกับรูปแบบข้อมูลได้ แต่ผลที่ได้ยังไม่เป็นฐานข้อมูลเชิงอุปนัยที่สมบูรณ์ เนื่องจากฟังก์ชันและชุดคำสั่งที่เพิ่มเข้ามายังเป็นลักษณะ loosely coupling แยกส่วนจาก DBMS นอกจากนี้ชุดคำสั่งที่ใช้ระบุเทคนิคการทำ mining ส่วนใหญ่ยังเป็นเพียงแนวคิดหรือโปรแกรมต้นแบบ นอกจากนี้วิธีการแทนและเก็บบันทึกรูปแบบข้อมูลยังมีวิธีการที่แตกต่างกันในแต่ละระบบ งานวิจัยนี้จึงได้เสนอแนวทางการออกแบบระบบจัดการฐานข้อมูลเชิงอุปนัย ที่ใช้มาตรฐานเดียวกันทั้งในส่วนการจัดการกับข้อมูลและส่วนจัดการกับรูปแบบข้อมูล

ลักษณะของฐานข้อมูลเชิงอุปนัย

งานวิจัยส่วนใหญ่ในสาขาการทำเหมืองข้อมูล มีมุมมองเกี่ยวกับฐานข้อมูลว่าเป็นเพียงแหล่งป้อนข้อมูลเข้าสู่ mining phase ของกระบวนการทำเหมืองข้อมูล จึงมักจะพัฒนา mining engine เป็นโมดูลที่แยกจากระบบจัดการฐานข้อมูล (database management system – DBMS) ในปี 1996 T. Imielinski และ H. Mannila [22] ได้เสนอแนวคิดของการปรับปรุง DBMS ให้เป็น KDDMS (knowledge and data discovery management system) เพื่อรองรับได้ทั้งงานด้านการจัดการฐานข้อมูลและการทำเหมืองข้อมูล ภาษาที่ใช้ในการสอบถามและจัดการกับข้อมูล จะต้องได้รับการปรับปรุงให้มีประสิทธิภาพมากกว่าภาษา SQL ที่ใช้ในฐานข้อมูลทั่วไป และโครงสร้างของข้อมูลในฐานข้อมูลจะมีความซับซ้อนขึ้นมากกว่าเป็นเพียงรูปแบบของเรคคอร์ด (หรือทูพเพิล) และรีเลชันที่ใช้อยู่ในฐานข้อมูลปกติ ฐานข้อมูลในรูปแบบใหม่นี้เรียกว่า *ฐานข้อมูลเชิงอุปนัย* (inductive databases) โดยสื่อความหมายถึงแหล่งรวมข้อมูลและรูปแบบข้อมูลที่ค้นพบหรืออุปนัย (induced) มาจากข้อมูล

หลังจากที่ H. Mannila ได้ร่วมเสนอแนวคิดพื้นฐานเกี่ยวกับฐานข้อมูลเชิงอุปนัย ในปีต่อมาเขาได้เสนอรายละเอียดเพิ่มเติมเกี่ยวกับโครงสร้าง และนิยามอย่างเป็นทางการของฐานข้อมูลเชิงอุปนัย [31, 32, 33] ดังนี้

<u>นิยามที่ 1</u> ฐานข้อมูลเชิงอุปนัย

ฐานข้อมูลเชิงอุปนัย คือ คู่ลำดับ (R, P) โดย R คือความสัมพันธ์หรือรีเลชันในฐานข้อมูล และ P คือแพทเทิร์นของความสัมพันธ์ในฐานข้อมูล โดย P จะอยู่ในรูปแบบของ (Q_R, e) เมื่อ Q_R คือแพทเทิร์นที่ได้จากการสอบถามข้อมูลในฐานข้อมูล และ e เป็นฟังก์ชันประเมินคุณสมบัติของแพทเทิร์น

ข้อมูล R และแพทเทิร์น P ในรูปที่ 1 แสดงตัวอย่างของฐานข้อมูลเชิงอุปนัย (ปรับปรุงจากตัวอย่างของ J.-F. Boulicaut และคณะ [6, 7]) แพทเทิร์น P สร้างขึ้นจากความสัมพันธ์ที่เป็นจริง (แทนด้วยค่า 1) ของแอททริบิวต์ X, Y, Z ในรีเลชัน R แพทเทิร์นในตัวอย่างนี้เป็นลักษณะของกฎความสัมพันธ์ (association rules) เช่น $X \Rightarrow Y$ แทนความสัมพันธ์ว่าเมื่อ X มีค่าเป็น 1 แล้ว Y มีค่าเป็น 1 โดยความสัมพันธ์นี้ปรากฏใน 1 เรคคอร์ดจากข้อมูลทั้งหมด 4 เรคคอร์ด จึงมีค่า support = 1/4 = 0.25 และ X มีค่าเป็น 1 จำนวน 3 เรค-คอร์ด แต่ในจำนวนนี้ Y มีค่าเป็น 1 เหมือน X เพียงเรคคอร์ดเดียว กฎนี้จึงมีความถูกต้อง หรือมีค่า confidence = 1/3 = 0.33 ดังนั้นในตัวอย่างนี้ $Q_R = \{ \text{LHS} \Rightarrow \text{RHS} \mid \text{LHS, RHS} \subseteq R \}$ และ e = (support, confidence)

R

X	Y	Z
1	0	0
1	1	1
1	0	1
0	1	1

P

pattern	support	confidence
X⇒Y	0.25	0.33
X⇒Z	0.50	0.66
Y⇒X	0.25	0.50
Y⇒Z	0.50	1.00
Z⇒X	0.50	0.66
Z⇒Y	0.50	0.66
XY⇒Z	0.25	1.00
XZ⇒Y	0.25	0.50
YZ⇒X	0.25	0.50

รูปที่ 1 ตัวอย่างของข้อมูลและรูปแบบข้อมูลในฐานข้อมูลเชิงอุปนัย

รูปแบบข้อมูลหรือแพทเทิร์น P ในรูปที่ 1 เกิดขึ้นจากการใช้คำสั่งเพื่อระบุเกณฑ์ต่างๆ และลักษณะของแพทเทิร์นที่ต้องการ การค้นหาแพทเทิร์นในฐานข้อมูลเชิงอุปนัยนิยามได้ดังต่อไปนี้

<u>นิยามที่ 2</u> การค้นหาแพทเทิร์นในฐานข้อมูลเชิงอุปนัย

กำหนดให้ r คือข้อมูล (instances) ในรีเลชัน R การค้นหาแพทเทิร์นจากคลาสของ L (แทน language ในรูปแบบที่ต้องการ เช่น กฎความสัมพันธ์) คือการค้นหาแพทเทิร์น p ที่ตรงตามเงื่อนไข q (r, p) และเรียกเซตของแพทเทิร์นที่ได้นี้ว่า ทฤษฎี (แทนด้วย Th)

$$\text{Th}(L, r, q) = \{ p \in L \mid q(r, p) \text{ is true} \}$$

จากนิยามข้างต้นเงื่อนไข q (r, p) คือ inductive query ที่ใช้ระบุการค้นหาแพทเทิร์น p ที่ผู้ใช้สนใจ จากมุมมองนี้การทำเหมืองข้อมูลจึงถูกพิจารณาว่าเป็นลักษณะหนึ่งของการสอบถาม (querying) ในกรอบคิดของฐานข้อมูลเชิงอุปนัย

งานวิจัยในช่วงระยะเวลาสิบปีที่ผ่านมาของการพัฒนาฐานข้อมูลเชิงอุปนัย แบ่งกลุ่มงานวิจัยได้เป็นสองกลุ่มใหญ่ [41] คือ กลุ่มแรกเน้นการค้นคว้าวิจัยในเชิงทฤษฎีเพื่อกำหนดรากฐานให้กับฐานข้อมูลเชิงอุปนัย โดยจะเน้นการกำหนดรูปแบบมาตรฐานของโครงสร้างข้อมูลในฐานข้อมูล และกำหนดคลาสมาตรฐานของ inductive queries งานวิจัยในกลุ่มที่สองซึ่งเป็นกลุ่มที่มีนักวิจัยเข้าร่วมเป็นจำนวนมาก จะเน้นในแง่มุมของการ

ใช้งานจริงโดยพยายามปรับปรุง DBMS ที่ใช้อยู่ในปัจจุบัน ให้สามารถทำเหมืองข้อมูลได้ด้วยการเพิ่มโมดูลต่าง ๆ เพื่อการทำ pattern mining และเพิ่มเติมชุดคำสั่ง SQL ให้สามารถสั่งการค้นหาแพทเทิร์นและจัดการกับแพทเทิร์น เช่น การบันทึก การแก้ไขเปลี่ยนแปลง และการสอบถาม

ในกลุ่มของนักวิจัยที่สนใจในแนวทางเชิงทฤษฎีพื้นฐานของฐานข้อมูลเชิงอุปนัยมี L. De Raedt และคณะ [14, 15] เป็นทีมงานหลักในด้านการออกแบบเชิงทฤษฎีโดยใช้ first-order logic เป็นคณิตศาสตร์พื้นฐานของแนวคิด นักวิจัยในกลุ่มนี้จะเน้นการออกแบบโมเดลของฐานข้อมูล [14] และกำหนดทฤษฎีของภาษาเพื่อการสอบถามข้อมูลและแพทเทิร์นจากฐานข้อมูล [15] รวมถึงการกำหนดรูปแบบพีชคณิตเพื่อการประมวลผล inductive queries [29] พีชคณิตนี้สามารถปรับปรุงจาก relational algebra ด้วยการเพิ่มเติมส่วน evaluation function เพื่อประเมินคุณภาพของแพทเทิร์น และส่วนของการปฏิบัติการกับแพทเทิร์น

งานวิจัยของกลุ่มที่เน้นการใช้งานจริงของฐานข้อมูลเชิงอุปนัย ซึ่งเป็นแนวทางที่ได้รับความสนใจจากนักวิจัยจำนวนมาก งานวิจัยส่วนใหญ่ในกลุ่มนี้มีแนวทางที่คล้ายกัน คือมุ่งเน้นไปที่การปรับปรุงภาษา SQL ในลักษณะของการต่อขยาย (SQL-extensions) ภาษาในลักษณะนี้ที่เกิดขึ้นในระยะแรกและได้รับการอ้างอิงถึงมาก ได้แก่ ภาษา DMQL (data mining query language) พัฒนาโดย J. Han และคณะ [21] และภาษาที่ใช้ชุดคำสั่ง MINE RULE พัฒนาโดย R. Meo และคณะ [34, 35, 36] ตัวอย่างในรูปที่ 2 (ดัดแปลงจากตัวอย่างใน [5]) แสดงคำสั่ง DMQL ระบุการค้นหาแพทเทิร์นประเภท characteristic rules จากฐานข้อมูล university_database และกำหนดลักษณะที่เกี่ยวข้อง คือ gpa, birth_place, grant ของนักศึกษาบัณฑิตศึกษาในสาขาคอมพิวเตอร์

```
use database university_database find characteristic rules
related to GPA, birth_place, grant, count (*)%
from students
where  status="graduate" and major="cs" with noise threshold=0.05
```

รูปที่ 2 ตัวอย่างคำสั่งค้นหารูปแบบข้อมูลของภาษา DMQL

การเพิ่มเติมคำสั่ง SQL ของ R. Meo และคณะ ใช้วิธีสร้างคำสั่ง MINE RULE เพื่อค้นหากฎความสัมพันธ์ (association rules) ดังตัวอย่างชุดคำสั่งในรูปที่ 3 (ดัดแปลงจากตัวอย่างใน [5]) ที่แสดงคำสั่งเพื่อใช้ค้นหาความสัมพันธ์ของสินค้า (item) จากรีเลชัน transaction(Date, CustID, Item, Value) โดยมีเงื่อนไขว่าสินค้านั้นจะต้องมีมูลค่าสูงกว่า 100 และลูกค้าซื้อสินค้าในคราวเดียวกันมากกว่า 4 ชิ้น

```
MINE RULE Associations AS
 SELECT DISTINCT 1..n Item AS BODY, 1..1 Item AS HEAD,SUPPORT,CONFIDENCE
 WHERE BODY.Value > 100 AND HEAD.Value > 100
 FROM transaction
 GROUP BY CustID HAVING COUNT(Item) > 4
 CLUSTER BY Date HAVING BODY.Date < HEAD.Date
EXTRACTING RULE WITH SUPPORT: 0.2, CONFIDENCE: 0.5
```

รูปที่ 3 ตัวอย่างคำสั่ง MINE RULE เพื่อการค้นหากฎความสัมพันธ์

ทั้งภาษา DMQL และ MINE RULE มีความสามารถในการค้นหาแพทเทิร์น แต่ยังไม่สามารถสอบถามแพทเทิร์น ความสามารถนี้ได้รับการพัฒนาเพิ่มเติมใน MSQL ที่พัฒนาโดย T. Imielinski และคณะ [23] ภาษา MSQL จำกัดการค้นหาแพทเทิร์นเฉพาะในประเภท association rules และในงานวิจัยต่อมาของทีมงานวิจัยจำนวนมาก [2, 8, 40, 45] ก็จำกัดความสนใจของการพัฒนาฐานข้อมูลเชิงอุปนัยเพื่อค้นหาเฉพาะแพทเทิร์นประเภทนี้เช่นเดียวกัน ทั้งนี้เนื่องจากเป็นแพทเทิร์นที่ใช้มากในงานทางด้านธุรกิจ

ใน DMX (data mining extensions) ของระบบจัดการฐานข้อมูล Microsoft SQL Server [42] นักพัฒนาระบบได้บรรจุคำสั่งในการค้นหาแพทเทิร์นประเภท classification แต่การทำงานกับแพทเทิร์นยังไม่มีขีดความสามารถในระดับการใช้คำสั่งซ้อนกัน นอกจากการขยายขีดความสามารถของ SQL ในฐานข้อมูลเชิงสัมพันธ์แล้ว ยังได้มีงานวิจัยที่พัฒนาภาษาในรูปแบบอื่น เช่น ODMQL (object data mining query language) [16] ที่ใช้กับฐานข้อมูลเชิงวัตถุ PMML (predictive model markup language) [38] ที่ใช้กับฐานข้อมูลเว็บ และจากความสนใจในการผนวกรวมฟังก์ชันการทำเหมืองข้อมูลเข้ากับชุดคำสั่ง SQL ทำให้มีการกำหนดมาตรฐานเป็น ISO SQL/MM [24] และกำหนดมาตรฐานส่วนเชื่อมต่อกับ Java [25]

ตั้งแต่ระยะแรกของการเกิดแนวคิดเกี่ยวกับฐานข้อมูลเชิงอุปนัย นอกจากการใช้ SQL เป็นพื้นฐานในการปรับปรุงเพิ่มเติมคำสั่งในการทำ pattern mining แล้ว ยังมีนักวิจัยในอีกกลุ่มหนึ่งที่ใช้ภาษาเชิงตรรกะเป็นพื้นฐานในการพัฒนาชุดคำสั่งเพื่อการค้นหาแพทเทิร์น นักวิจัยในกลุ่มนี้ ได้แก่ L. Dehaspe และคณะ [11,12] ที่พัฒนาคำสั่งและกระบวนการค้นหา frequent patterns โดยอาศัยพื้นฐานจากภาษา Datalog ซึ่งเป็นภาษาที่ใช้ทำงานกับฐานข้อมูลเชิงนิรนัย (deductive databases) แต่เนื่องจากข้อจำกัดของฐานข้อมูลเชิงนิรนัยที่จะมีประสิทธิภาพด้อยลงเมื่อข้อมูลมีปริมาณมาก C. Goh และคณะ [20] จึงได้เสนอวิธีการสุ่มเพื่อคัดเลือกเฉพาะข้อมูลตัวแทนมาใช้ในขั้นตอนการหา characteristic patterns การเพิ่มความสามารถด้านอุปนัยกับฐานข้อมูลเชิงนิรนัยได้รับความสนใจมาอย่างต่อเนื่อง โดยมีการนำเสนอแนวคิดทั้งในเชิงทฤษฎีพื้นฐาน [1, 3, 30, 39] และในด้านการออกแบบ query เพื่อค้นหาและจัดการกับแพทเทิร์น [13, 17, 19, 26, 44]

ในระยะหลังงานวิจัยด้านฐานข้อมูลเชิงอุปนัย ทั้งในกลุ่มที่พัฒนาทฤษฎีพื้นฐานและกลุ่มที่เน้นการประยุกต์ใช้งานจริง เริ่มลดความชัดเจนของการแบ่งแยกขอบเขตของงาน แนวทางของงานวิจัยเริ่มพัฒนาไปสู่การออกแบบระบบฐานข้อมูลเชิงอุปนัย IDBMS (inductive database management system) ที่มีทฤษฎีพื้นฐานรองรับและเพิ่มความสามารถให้เหมาะสมกับการใช้งานจริงโดยออกแบบ query ที่ใช้ในการค้นหาและจัดการกับแพทเทิร์นได้มากกว่าหนึ่งประเภท งานวิจัยในลักษณะการออกแบบระบบ IDBMS มีทั้งที่ใช้ first-order logic เป็นพื้นฐานของการออกแบบ [18] และที่เป็นลักษณะผสมผสานระหว่างวิธีการทางตรรกะที่มีการทำงานเชิงประกาศ (declarative) และวิธีการโปรแกรมเชิงสั่งงาน (imperative) โดยการเก็บข้อมูลและแพทเทิร์นจะอยู่ในฐานข้อมูลเชิงสัมพันธ์ ระบบในลักษณะผสมนี้ส่วนใหญ่เพิ่งจะเกิดขึ้นและขณะนี้อยู่ในระหว่างการพัฒนา ได้แก่ ระบบ CINQ [9], Psycho [10, 37, 43] และ ConQuest [4]

ในงานวิจัยนี้ ผู้วิจัยมีความสนใจในการพัฒนาระบบฐานข้อมูลเชิงอุปนัยที่ใช้ first-order logic เป็นพื้นฐานของการกำหนดกรอบแนวคิดและการออกแบบการทำงานของระบบ เนื่องจากวิธีการทางตรรกะนี้มีขีดความสามารถในเชิงประกาศสูงกว่าภาษา SQL เช่น คำสั่งดังตัวอย่างในรูปที่ 4 (ดัดแปลงจาก [19]) แสดงวิธีการระบุการค้นหาสินค้าที่ลูกค้านิยมซื้อคู่กันบ่อยๆ ของการซื้อของในคราวเดียวกัน การค้นหาสินค้ากระทำกับฐานข้อมูล transaction(date, customer, item, price, quantity) และระบุเงื่อนไขเพิ่มเติมว่าสินค้าที่ถูกซื้อคู่กันบ่อยนี้จะต้องถูกซื้อโดยลูกค้าจำนวนมากกว่า 10 คน

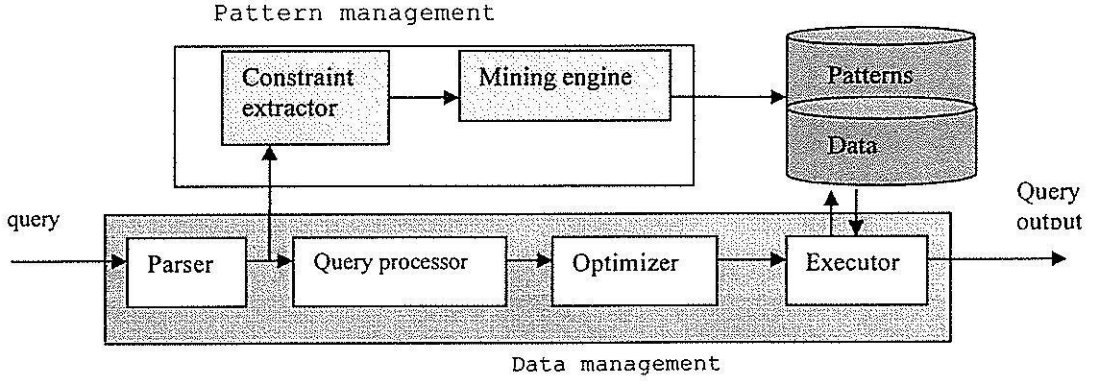
```
pair(I1,I2,count(C)):- transaction(D,C,I1,_,_),transaction(D,C,I2,_,_),I1\== I2.
ans(I1, I2) :- pair(I1, I2, C),  C > 10.
```

รูปที่ 4 ตัวอย่างคำสั่งค้นหา frequent pattern ในภาษาเชิงประกาศ

จากตัวอย่างคำสั่งในรูปที่ 4 จะเห็นได้ว่าลักษณะของการทำ pattern matching ในภาษาเชิงประกาศ ทำได้ง่ายกว่าในภาษาเชิงสั่งงาน ภาษาเชิงประกาศที่ใช้หลักการตรรกศาสตร์จึงเหมาะกับงาน pattern mining นอกจากความสามารถในเชิงประกาศแล้ว first-order logic ยังเหมาะสมสำหรับการพัฒนาการทำเหมืองข้อมูลไปสู่ความสามารถในด้านการค้นหาแพทเทิร์นที่มีการระบุเงื่อนไข (constraint pattern mining) และการค้นหาแพทเทิร์นจากหลายความสัมพันธ์หรือหลายฐานข้อมูลในลักษณะ multi-relation mining

กรอบแนวคิดของการออกแบบฐานข้อมูลเชิงอุปนัย

ในการออกแบบฐานข้อมูลเชิงอุปนัย นอกจากส่วนของการค้นหาและจัดการกับแพทเทิร์นแล้ว ผู้วิจัยยังได้เสนอกรอบแนวคิด (framework) เกี่ยวกับการออกแบบและพัฒนา IDBMS เพิ่มเติมจากที่นักวิจัยต่าง ๆ ได้เสนอไว้แล้ว คือ ให้มีการเพิ่มส่วนเชื่อมโยงของการใช้ประโยชน์จากแพทเทิร์น ป้อนกลับไปยังส่วนประมวลผลข้อคำถาม เพื่อเพิ่มประสิทธิภาพของกระบวนการ query answering และ semantic query optimization [27, 28] องค์ความรู้ใหม่ที่ได้จากงานวิจัยนี้จะช่วยให้สามารถพัฒนาระบบฐานข้อมูลไปสู่ความสามารถทั้งในเชิงอุปนัยและนิรนัยได้สมบูรณ์มากขึ้น กรอบแนวคิดของการออกแบบฐานข้อมูลเชิงอุปนัยแสดงเป็นแผนภาพได้ดังรูปที่ 5



รูปที่ 5 กรอบแนวคิดแสดงส่วนประกอบของระบบจัดการฐานข้อมูลเชิงอุนัย

กรอบแนวคิดของระบบจัดการฐานข้อมูลเชิงอุปนัย (inductive database management system -- IDBMS) ประกอบด้วยส่วนประกอบหลักสองส่วน คือ ส่วนจัดการกับข้อมูล (data management) และส่วนจัดการกับรูปแบบข้อมูล (pattern management) ในส่วนที่จัดการกับรูปแบบข้อมูลมีการออกแบบโครงสร้างมาตรฐานของการแทนข้อมูลและรูปแบบข้อมูล (data and pattern representation) เนื่องจากในปัจจุบันรูปแบบข้อมูลมีได้หลายลักษณะ เช่น generalized rule-based patterns, clustering patterns, association patterns, time-series patterns การกำหนดรูปแบบให้เป็นมาตรฐานเดียวกันเป็นสิ่งจำเป็นสำหรับระบบปิด เพื่อประโยชน์ในการเรียกใช้รูปแบบข้อมูลซ้อนกัน (nested patterns) รวมถึงให้ query สามารถทำงานกับรูปแบบข้อมูลใหม่ ๆ ที่จะเพิ่มเติมขึ้นมาในอนาคตได้

ส่วนจัดการกับข้อมูลได้มีการออกแบบและพัฒนา operators ในการประมวลผล query โดยรูปแบบของ query พัฒนาเพิ่มเติมจาก query ที่ใช้ในภาษา Datalog ซึ่งมีทฤษฎีพื้นฐานจาก first-order logic ชุดคำสั่งที่ออกแบบนี้จะสอบถามได้ทั้งข้อมูลและรูปแบบข้อมูล โดยในงานวิจัยนี้จะพิจารณารูปแบบข้อมูลใน 3 กลุ่มหลัก คือ classification patterns, clustering patterns และ association patterns ในส่วนของระบบการจัดการกับรูปแบบข้อมูล ได้รับการออกแบบให้สามารถทำการค้นหารูปแบบข้อมูลแบบกำหนดเงื่อนไข (constraint pattern mining) การออกแบบวิธีการ update ข้อมูลและรูปแบบข้อมูล จะพิจารณาวิธีการปรับปรุงรูปแบบข้อมูลในแนวทางของ incremental mining ซึ่งจะให้ประสิทธิภาพที่ดีกว่าในแบบ batch

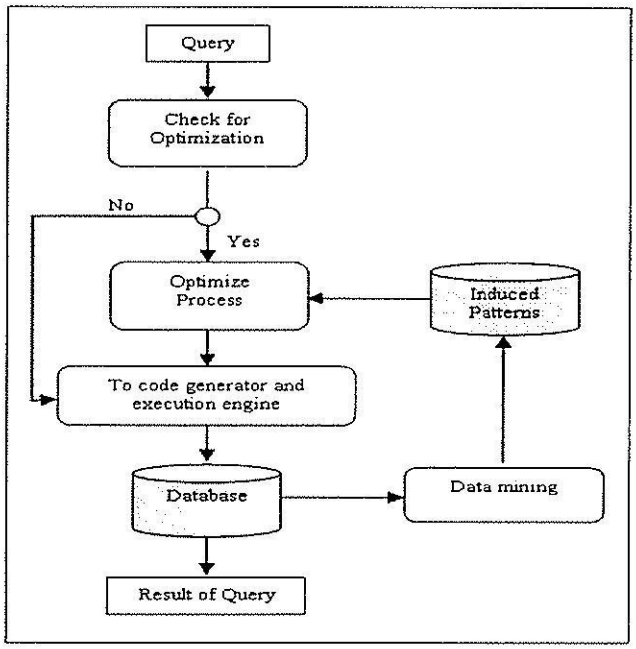
การทดสอบความสามารถของฐานข้อมูลเชิงอุปนัย

การออกแบบระบบฐานข้อมูลเชิงอุปนัยของงานวิจัยนี้ มีจุดมุ่งหมายหลักที่จะให้สามารถนำรูปแบบข้อมูลมาใช้เป็นฐานความรู้เพื่อการปรับปรุงการตอบข้อคำถาม (query optimization) ทำได้ดีขึ้น ขั้นตอนการปรับปรุงข้อคำถามแสดงได้ดังรูปที่ 6 การทดสอบประสิทธิภาพการตอบข้อคำถาม ใช้ข้อมูล customers ซึ่งเป็นข้อมูลสังเคราะห์ มีโครงสร้าง (schema) ดังนี้

customers (customerID, name, address, city, country, birthdate, marital, gender, education, member_card, total_children, occupation, houseowner)

และรูปแบบข้อมูลในลักษณะของ association rules ที่ค้นพบได้จากข้อมูล customers แสดงได้ดังนี้

```
gender=m                ⇒  marital=s
total_children=0        ⇒  marital=s
total_children=0        ⇒  gender=m
gender=m                ⇒  total_children=0
houseowner=no           ⇒  marital=s
member_card=bronze      ⇒  occupation=skilled_manual
marital=m               ⇒  gender=f
marital=m               ⇒  houseowner=yes
city=los_angeles        ⇒  houseowner=yes
city=nation_city        ⇒  occupation=skilled_manual
```



รูปที่ 6 ขั้นตอนการปรับปรุงการตอบข้อคำถามของระบบจัดการฐานข้อมูลเชิงอุนัย

รูปแบบข้อมูลเหล่านี้ถูกนำไปใช้ในการแปลงรูปแบบข้อคำถาม (query rewriting) โดยใช้ตัวอย่างข้อคำถาม Q1 ถึง Q5 สอบถามข้อมูลลูกค้าด้วยเงื่อนไขต่าง ๆ ผลการทดสอบแสดงการปรับปรุงข้อคำถาม (Q1', Q2', Q3', Q4', Q5') ด้วยรูปแบบข้อมูล (pattern) และแสดงเวลาที่ใช้ในการตอบแต่ละข้อคำถาม

Q1: SELECT * FROM customers WHERE city='santa cruz' AND gender='f';

Pattern: *city=santa_cruz ⇒ gender=m*

Q1': None: detection of unsatisfiable condition
Answer: null

Time(ms):

query	Test#1	Test#2	Test#3	Test#4	Test#5
Q1	50	50	51	50	51
Q1'	0	0	0	0	0

Q2: SELECT * FROM customers WHERE city='santa cruz' AND gender='m' AND marital = 'm';

Pattern: *city=santa_cruz ⇒ gender=m, marital=s*

Q2': None: detection of unsatisfiable condition
Answer: null

Time(ms):

query	Test#1	Test#2	Test#3	Test#4	Test#5
Q2	109	103	104	101	104
Q2'	0	0	0	0	0

```
Q3:   SELECT  * FROM customers WHERE city='los angeles' AND houseowner='yes';
      Pattern:  city=los_angeles ⇒ houseowner=yes
Q3':      SELECT  *  FROM customers  WHERE city = 'los angeles';
Answer:   Q3 = 27,660 tuples; Q3'= 27,660 tuples
```

Time(ms):

query	Test#1	Test#2	Test#3	Test#4	Test#5
Q3	1336	1442	1406	1429	1422
Q3'	1126	1303	1268	1273	1230

```
Q4:   SELECT  *  FROM customers WHERE gender='m' AND marital='s'
                                 AND total_children = '0';
      Pattern:  gender = m ⇒ marital= s, total_children = 0
Q4':      SELECT  *  FROM customers WHERE gender = 'm';
Answer:   Q4 = 71,916 tuples; Q4'= 71,916 tuples
```

Time(ms):

query	Test#1	Test#2	Test#3	Test#4	Test#5
Q4	4559	4043	4665	4043	4440
Q4'	3377	3717	3489	3690	3404

```
Q5:  SELECT  *  FROM customers  WHERE city='santa cruz' AND gender='m'
                                AND member_card = 'bronze';
     Pattern:  city = santa cruz ⇒ gender = m
Q5':      SELECT * FROM customers WHERE gender='m' AND member_card='bronze';
Answer:   Q5 = 16,596 tuples; Q5'= 16,596 tuples
```

Time(ms):

query	Test#1	Test#2	Test#3	Test#4	Test#5
Q5	1313	1908	1185	1749	1375
Q5'	936	1274	1074	962	951

จากผลการทดสอบจะเห็นได้ว่า รูปแบบข้อมูลช่วยค้นพบความขัดแย้งในเงื่อนไขของข้อคำถาม Q1 และ Q2 ทำให้สามารถตอบได้ทันทีว่าไม่มีคำตอบ (null answer) ในขณะที่ข้อคำถาม Q3, Q4, Q5 สามารถถูกปรับปรุงให้มีเงื่อนไขลดลงได้ด้วยรูปแบบข้อมูลที่ค้นพบ ทำให้ใช้เวลาลดลงในการตอบข้อคำถามเหล่านั้น

สรุปและข้อเสนอแนะ

งานวิจัยนี้นำเสนอแนวทางในการขยายขีดความสามารถของระบบฐานข้อมูลที่ใช้งานอยู่ในปัจจุบัน ให้มีขีดความสามารถเพิ่มขึ้นโดยผนวกฟังก์ชันของการทำเหมืองข้อมูล การเพิ่มฟังก์ชันนี้จะมีประโยชน์ในการค้นหาความสัมพันธ์ในลักษณะของ association rules, functional dependencies, semantic constraints และความสัมพันธ์

ในรูปแบบอื่น ๆ จากข้อมูลที่เก็บไว้ในฐานข้อมูล ความสัมพันธ์ที่ค้นพบนี้จะเรียกว่ารูปแบบข้อมูล หรือ แพทเทิร์นของข้อมูล ฐานข้อมูลที่บันทึกทั้งข้อมูลและรูปแบบข้อมูลนี้เรียกว่า ฐานข้อมูลเชิงอุปนัย

ในการออกแบบระบบจัดการฐานข้อมูลเชิงอุปนัย จึงต้องมีทั้งส่วนจัดการกับข้อมูล และ ส่วนจัดการกับรูปแบบข้อมูล ในงานวิจัยนี้เน้นการออกแบบฐานข้อมูลเพื่อให้สามารถค้นหารูปแบบข้อมูลจากฐานข้อมูล และใช้รูปแบบข้อมูลเพิ่มประสิทธิภาพการตอบข้อคำถาม โดยผลการทดลองเบื้องต้นยืนยันข้อสมมุติฐานว่าแนวความคิดนี้สามารถนำไปใช้ประโยชน์ได้จริง การพัฒนางานวิจัยนี้ต่อไปในอนาคตจึงเป็นแนวทางของการขยายขอบเขตงานวิจัยให้ครอบคลุมความสามารถอื่นของระบบฐานข้อมูล เช่น การจัดการทรานแซคชัน การ update ข้อมูล และการใช้งานจริงกับฐานข้อมูลขนาดใหญ่

กิตติกรรมประกาศ

งานวิจัยนี้ได้รับการสนับสนุนงบประมาณจากสำนักงานคณะกรรมการวิจัยแห่งชาติ สำนักงานคณะกรรมการการอุดมศึกษาและสำนักงานกองทุนสนับสนุนการวิจัย (สกว., รหัสโครงการ RMU-5080026) หน่วยวิจัยด้านวิศวกรรมข้อมูลและการค้นหาความรู้ เป็นหน่วยปฏิบัติการวิจัยที่ได้รับการสนับสนุนการดำเนินงานและงบประมาณบางส่วนจากมหาวิทยาลัยเทคโนโลยีสุรนารี

เอกสารอ้างอิง

[1] M. Aragao and F. Fernandes. Logic-based integration of query answering and knowledge discovery. *Proc. 6th Int. Conf. on Flexible Query Answering Systems*, pp. 68-83, 2004.

[2] M. Botta, J.-F. Boulicaut, C. Masson, and R. Meo. A comparison between query languages for the extraction of association rules. *Proc. 3rd Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK'02)*, pp. 1-10, 2002.

[3] I. Bartolini, P. Ciaccia, I. Ntoutsi, M. Patella, and Y. Theodoridis. A unified and flexible framework for comparing simple and complex patterns. *Proc. 8th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'04)*, pp. 496-499, 2004.

[4] F. Bonchi, F. Giannotti, C. Lucchese, S. Orlando, R. Perego, and R. Trasarti. ConQuest: A constraint-based querying system for exploratory pattern discovery. *Proc. IEEE Int. Conf. on Data Engineering (ICDE'06)*, pp. 159-160, 2006.

[5] F. Bonchi, F. Giannotti, C. Lucchese, S. Orlando, R. Perego, and R. Trasarti. On interactive pattern mining from relational databases. *Proc. 5th Int. Workshop on Knowledge Discovery in Inductive Databases (KDID'06)*, pp. 42-62, 2006.

[6] J.-F. Boulicaut, M. Klemettinen, and H. Mannila. Querying inductive databases: A case study on the MINE RULE operator. *Proc. 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PAKDD'98)*, pp. 194-202, 1998.

[7] J.-F. Boulicaut, M. Klemettinen, and H. Mannila. Modeling KDD processes within the inductive database framework. *Proc. 1st Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK'99)*, pp. 293-302, 1999.

[8] T. Calders, B. Goetals, and A. Prado. Integrating pattern mining in relational databases. *Proc. 10th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'06)*, pp. 454-461, 2006.

[9] CINQ project, 2003. http://www.cinq-project.org.

[10] B. Catania, A. Maddalena, and M. Mazza. Psycho: A prototype system for pattern management. *Proc. 31st Int. Conf. on Very Large Data Bases (VLDB'05)*, pp. 1346-1349, 2005.

[11] L. Dehaspe. Frequent pattern discovery in first order logic. *PhD Thesis*, Katholieke Universiteit Leuven, 1998.

[12] L. Dehaspe and H. Toivonen. Discovery of frequent datalog patterns. *Data Mining and Knowledge Discovery*, 3(1):7-36, 1999.

[13] L. De Raedt. A logical database mining language. *Proc. 10th Int. Conf. on Inductive Logic Programming*, pp. 78-92, 2000.

[14] L. De Raedt. A perspective on inductive databases. *SIGKDD Explorations*, 4(2):69-77, 2003.

[15] L. De Raedt M. Jaeger, S. Lee, and H. Mannila. A theory of inductive query answering. *Proc. IEEE Int. Conf. on Data Mining (ICDM'02)*, pp. 123-130, 2002.

[16] M. Elfeky, A. Saad, and S. Fouad. ODMQL: Object data mining query language. *Proc. Int. Symposium on Objects and Databases*, pp. 128-140, 2000.

[17] F. Giannott and G. Manco. Querying inductive databases via logic-based user-defined aggregates. *Proc. 3rd European Conf. on Principles of Data Mining and Knowledge Discovery (PKDD'99)*, pp. 125-135, 1999.

[18] F. Giannott, G. Manco, D. Pedreschi, and F. Turini. Experiences with a logic-based knowledge discovery support environment. *Proc. 6th Italian Congress of Artificial Intelligence*, pp. 202-213, 2000.

[19] F. Giannott, G. Manco, and F. Turini. Specifying mining algorithms with iterative user-defined aggregates. *IEEE Transactions on Knowledge and Data Engineering*, 16(10):1232-1246, 2004.

[20] C. Goh, M. Tsukamoto, and S. Nishio. Knowledge discovery in deductive databases with large deduction results: The first step. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):952-956, 1996.

[21] J. Han, Y. Fu, W. Wang, K. Koperski, and O. Zaiane. DMQL: A data mining query language for relational databases. *Proc. ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pp. 27-34, 1996.

[22] T. Imielinski and H.Mannila. A database perspective on knowledge discovery. *Communications of the ACM*, 39(11):58-46, 1996.

[23] T. Imielinski and V. Virmani. MSQL: A query language for database mining. *Data Mining and Knowledge Discovery*, 3(4):373-408, 1998.

[24] ISO SQL/MM Part 6, 2001. http://www.sql-99.org/sc32/WG4/Progression_Documents/FCD/fcd-datamining-2001-05.pdf.

[25] Java Data Mining API, 2003. http://www.jcp.org/jsr/detail/73.prt.

[26] B. Jeudy and J.-F. Boulicaut, Constraint-based discovery and inductive queries: Application to association rule mining. *Proc. ESF Exploratory Workshop on Pattern Detection and Discovery*, pp. 110-124, 2002.

[27] N. Kerdprasop and K. Kerdprasop. Semantic knowledge integration to support inductive query optimization. *Proc. 8th Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK'07)*, pp. 157-169, 2007.

[28] K. Kerdprasop, N. Kerdprasop, and A. Ritthongchailert. Query answering in relational inductive databases. *Proc. 18th Int. Workshop on Database and Expert Systems Applications (DEXA'07)*, pp. 329-333, 2007.

[29] S. Lee and L. De Raedt. An algebra for inductive query evaluation. *Proc. IEEE Int. Conf. on Data Mining (ICDM'03)*, pp. 147-154, 2003.

[30] G. Manco. Foundations of a logic-based framework for intelligent data analysis. *PhD Thesis*, University of Pisa, 2001.

[31] H. Mannila. Inductive databases and condensed representations for data mining. *Proc. Int. Symposium on Logic Programming*, pp. 21-30, 1997.

[32] H. Mannila. Theoretical frameworks for data mining. *SIGKDD Explorations*, 1(2):30-32, 2000.

[33] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241-258, 1997.

[34] R. Meo, G. Psaila, and S. Ceri. A new SQL-like operator for mining association rules. *Proc. 23rd Int. Conf. on Very Large Data Bases (VLDB'96)*, pp. 122-133, 1996.

[35] R. Meo, G. Psaila, and S. Ceri. A tightly-coupled architecture for data mining. *Proc. IEEE Int. Conf. on Data Engineering (ICDE'98)*, pp. 316-322, 1998.

[36] R. Meo, G. Psaila, and S. Ceri. An extension to SQL for mining association rules. *Data Mining and Knowledge Discovery*, 2(2):195-224, 1998.

[37] PANDA project, 2002. http://dke.cti.gr/panda.

[38] Predictive Model Markup Language (PMML), 2003. http://www.dmg.org/pmmlspecs_v2/pmml_v2_0.html

[39] S. Rizzi, E. Bertino, B. Catania, M. Golfarelli, M. Halkidi, M. Terrovitis, P. Vassiliadis, M. Vazirgiannis, and E. Vrachnos. Towards a logical model for patterns. *Proc. 22nd Int. Conf. on Conceptual Modeling (ER'03)*, pp. 77-90, 2003.

[40] S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications. *Data Mining and Knowledge Discovery*, 4(2-3):89-125, 2000.

[41] A. Siebes. Data mining in inductive databases. *Proc. 4th Int. Workshop on Knowledge Discovery in Inductive Databases (KDID'05)*, pp. 1-23, 2005.

[42] Z. Tang and J. MacLennan. *Data Mining with SQL Server 2005*. John Wiley & Sons, 2005.

[43] M. Terrovitis, P. Vassiliadis, S. Skiadopoulos, E. Bertino, B. Catania, A. Maddalena, and S. Rizzi. Modeling and language support for the management of pattern-bases. *Data and Knowledge Engineering*, 62(2):368-397, 2007.

[44] I. Toroslu and M. Yetisgen-Yildiz. Data mining in deductive databases using query flocks. *Expert Systems with Applications*, 28(3):395-407, 2005.

[45] H. Wang and C. Zaniolo. ATLaS: A native extension of SQL for data mining. *Proc. 3rd SIAM Int. Conf. on Data Mining*, pp. 130-144, 2003.

กรอบแนวคิดสำหรับระบบผู้เชี่ยวชาญแบบกฎเชิงอุปนัย

A Framework for Inductive Rule-Based Expert Systems

กิตติศักดิ์ เกิดประสพ นิตยา เกิดประสพ และ เอกสิทธิ์ เพชรดี
หน่วยปฏิบัติการวิจัยด้านวิศวกรรมข้อมูลและการค้นหาความรู้ สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีสุรนารี 111 ถนนมหาวิทยาลัย ต.สุรนารี อ.เมือง จ.นครราชสีมา 30000
โทรศัพท์ 0 4422 4349 โทรสาร 0 4422 4220 E-mail: kerdpras@ccs.sut.ac.th

บทคัดย่อ

บทความนี้นำเสนอความก้าวหน้าในการออกแบบและพัฒนาระบบผู้เชี่ยวชาญแบบกฎ ในรูปแบบใหม่ที่นอกจากจะมีกฎที่ถูกสร้างเตรียมไว้ล่วงหน้าแล้ว ระบบยังสามารถสังเคราะห์กฎใหม่เพิ่มเติมได้จากแหล่งข้อมูลที่มีอยู่ กระบวนการสังเคราะห์กฎใหม่ทำได้ด้วยการใช้เทคนิคการค้นหาความรู้ แต่เนื่องจากความรู้ที่สังเคราะห์ได้มักจะมีปริมาณมากเกินไปและความรู้บางส่วนเป็นความรู้ที่ไม่เกี่ยวข้อง งานวิจัยนี้จึงได้ออกแบบระบบให้มีโอเปอเรเตอร์ที่ช่วยในการกรองความรู้เพื่อคัดเลือกไว้เฉพาะความรู้ในรูปแบบของกฎที่สามารถใช้ประโยชน์ ความรู้ที่สังเคราะห์และคัดเลือกแล้วรวมทั้งกฎที่ออกแบบไว้ล่วงหน้าจะถูกนำมาประกอบกันเพื่อทำหน้าที่เป็นฐานความรู้ของระบบผู้เชี่ยวชาญ

คำสำคัญ : ระบบผู้เชี่ยวชาญ, การสังเคราะห์ความรู้, การกรองความรู้

Abstract

This article presents work in progress on a new methodology for the design and implementation of the next generation rule-based expert systems. In addition to the set of predefined rules, we include the rules that are automatically induced from the data repositories. The inductive process has been done through the knowledge discovery techniques. The induced knowledge is normally big and sometimes irrelevant, therefore, we propose a filter operator for useful-rule selection. The induced, as well as predefined, rules together form a knowledge base for the inductive expert system.

Keywords: expert system, knowledge induction, knowledge filtration

1. บทนำ

ในช่วงศตวรรษที่ 20 วงการคอมพิวเตอร์มีความตื่นตัวอย่างมากในงานด้านปัญญาประดิษฐ์ (Artificial Intelligence, AI) ที่พยายามจะโปรแกรมให้เครื่องคอมพิวเตอร์สามารถคิดแก้ปัญหาได้เองเหมือนมนุษย์ ในปัจจุบันนักคอมพิวเตอร์ยังไม่สามารถบรรลุวัตถุประสงค์ที่ได้ตั้งไว้ แต่ผลพลอยได้จากความพยายามนี้ คือ ความก้าวหน้าในสาขาย่อยต่างๆ ของปัญญาประดิษฐ์ ได้แก่ สาขาหุ่นยนต์ (robotics), สาขาการเรียนรู้ของเครื่องจักร (machine learning), สาขาการประมวลผลภาษาธรรมชาติ (natural language processing) และสาขาอื่นๆ อีกมากรวมถึงสาขาผู้เชี่ยวชาญ (expert systems)

ระบบผู้เชี่ยวชาญเป็นระบบที่ประกอบด้วยฐานความรู้ (knowledge base) และโปรแกรมที่เรียกว่าเครื่องอนุมาน (inference engine) ระบบผู้เชี่ยวชาญทำหน้าที่รวบรวมความรู้จากผู้เชี่ยวชาญไว้ในฐานความรู้เพื่อใช้ความรู้นั้นแนะนำวิธีแก้ปัญหาในเรื่องต่างๆ ข้อแนะนำมักจะแสดงในรูปของกฎ IF <condition> THEN <action> เช่น ตัวอย่างต่อไปนี้แสดงข้อแนะนำอย่างง่ายในการแก้ปัญหาเครื่องยนต์สตาร์ทไม่ติด

IF car won't start THEN check battery.

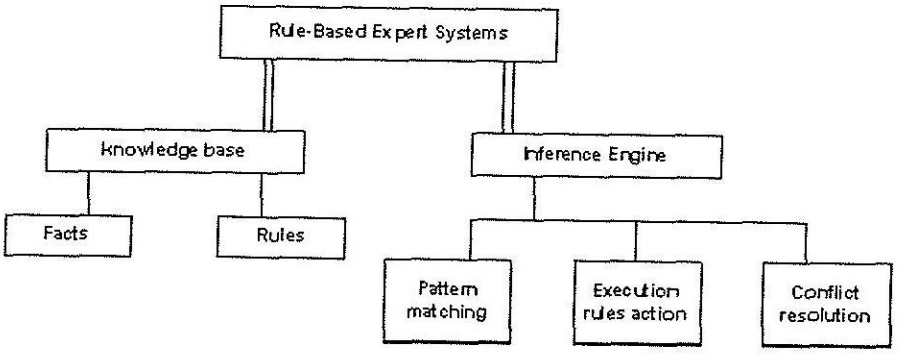
IF car won't start THEN check gas.

IF check battery AND battery bad THEN replace battery.

IF check gas AND no gas THEN fill gas tank.

ส่วนประกอบหลักของระบบผู้เชี่ยวชาญประกอบด้วยฐานความรู้และเครื่องอนุมาน โดยเครื่องอนุมานจะประกอบด้วยโปรแกรมย่อยอีกอย่างน้อย 3 โปรแกรม (ดังรูปที่ 1) คือโปรแกรมที่ทำหน้าที่จับคู่รูปแบบส่วนเงื่อนไขของกฎ (pattern matching), โปรแกรมดำเนินงานตามที่ระบุไว้ในส่วนการกระทำของกฎ (execution of the action past) และโปรแกรมช่วยแก้ปัญหาเมื่อกฎขัดแย้งกันเอง (conflict resolution)

การจัดโครงสร้างของระบบผู้เชี่ยวชาญให้ส่วนฐานความรู้ซึ่งเป็นส่วนเก็บข้อมูล แยกออกจากส่วนอนุมานซึ่งเป็นส่วนโปรแกรม ช่วยให้การพัฒนาระบบผู้เชี่ยวชาญระบบแรกที่ใช้โครงสร้างนี้คือ MYCIN[1] ซึ่งเป็นระบบผู้เชี่ยวชาญที่ให้คำแนะนำในการวินิจฉัยโรค จากความสำเร็จของ MYCIN ทำให้เกิดระบบผู้เชี่ยวชาญที่ใช้โครงสร้างแบบเดียวกันนี้อีกเป็นจำนวนมาก เช่น DENDRAL ใช้ช่วยวิเคราะห์โครงสร้างทางเคมี, DIPMETER และ PROSPECTOR ใช้ช่วยวิเคราะห์ข้อมูลทางธรณีวิทยาเพื่อแนะนำการขุดเจาะหาน้ำมันและแหล่งแร่ ระบบผู้เชี่ยวชาญเหล่านี้และระบบอื่นๆ ที่เกิดขึ้นในช่วงทศวรรษที่ 1970 และ 1980 จัดเป็นระบบผู้เชี่ยวชาญรุ่นที่1 [2]



รูปที่ 1 ส่วนประกอบหลักของระบบผู้เชี่ยวชาญแบบกฎ

ระบบผู้เชี่ยวชาญรุ่นที่ 1 สร้างฐานความรู้ด้วยวิธีให้วิศวกรความรู้ (knowledge engineer) ทำหน้าที่สัมภาษณ์ผู้เชี่ยวชาญเพื่อรวบรวมความรู้ แล้วแปลงรูปแบบความรู้เหล่านั้นให้อยู่ในรูปแบบของกฎ IF-THEN ขั้นตอนนี้เรียกว่าการแสวงหาความรู้ (knowledge acquisition) ซึ่งเป็นขั้นตอนที่ต้องใช้เวลามาก จึงมีความพยายามปรับปรุงขั้นตอนนี้ให้เป็นกระบวนการอัตโนมัติมากขึ้นเพื่อให้สามารถสร้างฐานความรู้ได้รวดเร็วขึ้นทำให้เกิดพัฒนาการเป็นระบบผู้เชี่ยวชาญรุ่นที่ 2 [3]

ลักษณะเด่นของระบบผู้เชี่ยวชาญรุ่นที่ 2 คือมีการประยุกต์ใช้เทคนิคจากสาขาการเรียนรู้ของเครื่องจักรมาใช้ช่วยในการสร้างความรู้ที่อยู่ในรูปแบบของกฎ ซึ่งเป็นขั้นตอนการแสวงหาความรู้และเรียกระบบผู้เชี่ยวชาญแบบนี้ว่าระบบผู้เชี่ยวชาญเชิงอุปนัย (Inductive expert system) [4-10]

ในงานวิจัยนี้คณะผู้วิจัยเสนอกรอบแนวคิดเพิ่มเติมจากระบบผู้เชี่ยวชาญเชิงอุปนัยที่มีผู้ใช้อยู่ในปัจจุบันด้วยการประยุกต์ใช้เทคนิคการเรียนรู้ของเครื่องจักรโดยเน้นที่การเรียนรู้เชิงอุปนัย (inductive learning) มาช่วยทั้งในขั้นตอนของการแสวงหาความรู้และขั้นตอนการเรียนรู้แพทเทิร์นการเลือกกฎของเครื่องอนุมาน เพื่อนำมาใช้ช่วยในการจัด

ฐานความรู้ (knowledge reorganization) เพื่อลดปัญหาความขัดแย้งของกฎ และช่วยหาฮิวริสติก (heuristics) เพื่อเพิ่มความเร็วในการทำงานของขั้นตอนการอนุมานความรู้

2. ระบบผู้เชี่ยวชาญเชิงอุปนัย

ฐานความรู้ที่สร้างจากการสัมภาษณ์ผู้เชี่ยวชาญ แล้วแปลงเป็นความรู้ในรูปแบบของกฎจะมีลักษณะดังต่อไปนี้ (ใช้ตัวอย่างจาก [8])

Rule1: IF Grade is Director THEN decision is Pass.

Rule2: IF Grade is Senior Manager AND Hotel is A THEN decision is Reject.

Rule3: IF Grade is Senior Manager AND Hotel is B AND Department is Accounts THEN decision is Pass.

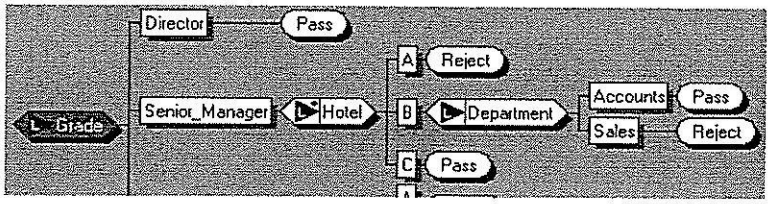
Rule4: IF Grade is Senior Manager AND Hotel is B AND Department is Sales THEN decision is Reject.

Rule5: IF Grade is Senior Manager AND Hotel is C THEN decision is Pass.

Rule6: IF Grade Junior Manager AND Hotel is A or B THEN decision is Reject.

Rule7: IF Grade is Junior Manager AND Hotel is C THEN decision is Pass.

ตัวอย่างความรู้ข้างต้นเป็นกฎที่บริษัทใช้ในการตัดสินใจว่าการออกไปปฏิบัติงานภายนอกสำนักงานของหน่วยงานจะสามารถเบิกค่าที่พักได้หรือไม่ ในระบบผู้เชี่ยวชาญเชิงอุปนัยระบบสามารถสังเคราะห์กฎเช่นนี้ได้โดยอัตโนมัติจากข้อมูลที่เป็นกรณีตัวอย่างของการตัดสินใจเบิกหรือไม่ให้เบิก ข้อมูลเหล่านี้เป็นข้อมูลในอดีตที่รวบรวมไว้ในฐานข้อมูล และสามารถสังเคราะห์เป็นรูปแบบสรุปเพื่อใช้เป็นความรู้ประกอบการตัดสินใจ ถ้าอัลกอริทึมการสังเคราะห์ความรู้เป็นแบบการอุปนัยด้วยต้นไม้ตัดสินใจ (decision-tree induction) ความรู้ที่ได้จะเป็นโครงสร้างต้นไม้ตัดสินใจ ดังรูปที่ 2



รูปที่ 2 ความรู้ในรูปแบบของต้นไม้ตัดสินใจ

3. โครงสร้างของระบบผู้เชี่ยวชาญเชิงอุปนัย

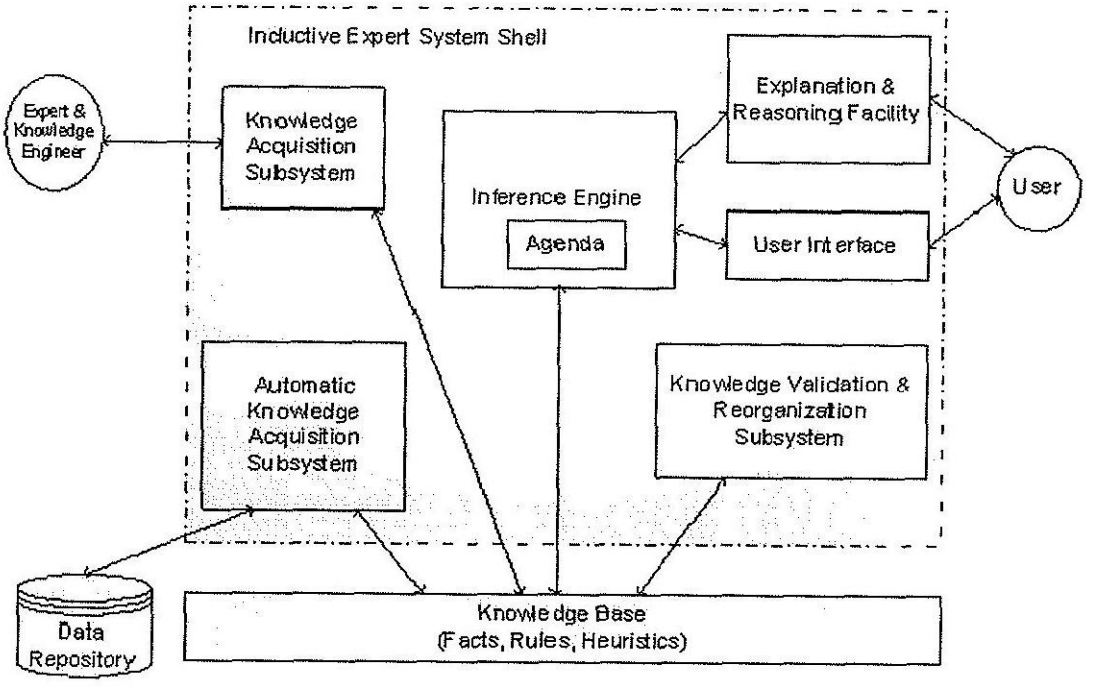
ระบบผู้เชี่ยวชาญเชิงอุปนัยที่ผสานเทคนิคการสังเคราะห์ความรู้ เพื่อให้ขั้นตอนการแสวงหาความรู้ทำได้โดยอัตโนมัติ มีโครงสร้างของระบบดังแสดงในรูปที่ 3

ส่วนประกอบต่างๆ ของระบบผู้เชี่ยวชาญเชิงอุปนัยมีหน้าที่ดังนี้

- Knowledge Acquisition Subsystem เป็นโปรแกรมช่วยงานเพื่อให้ผู้เชี่ยวชาญและวิศวกรความรู้สามารถสร้างความรู้ในรูปแบบของกฎ หรือรูปแบบอื่นๆ เก็บไว้ในฐานความรู้
- Automatic Knowledge Acquisition Subsystem เป็นโปรแกรมที่ทำหน้าที่สังเคราะห์ความรู้ในรูปแบบของกฎเพื่อการตัดสินใจ (decision rules) และกฎแสดงความสัมพันธ์ (association rules)
- Inference Engine เป็นส่วนประกอบหลักของระบบผู้เชี่ยวชาญทำหน้าที่อนุมานจากเหตุการณ์หรือจากเงื่อนไขที่เกิดขึ้น เพื่อเลือกกฎที่เกี่ยวข้องแสดงเป็นข้อแนะนำให้แก่ผู้ใช้ระบบ ในกรณีที่มีกฎที่เกี่ยวข้องหลายกฎเครื่องอนุมานจะบรรจุกฎเหล่านั้นไว้ในส่วน Agenda เพื่อจัดลำดับความสำคัญและดำเนินงานกับกฎที่มีความสำคัญ

สูงสุดก่อน ซึ่งแพทเทิร์นการเลือกกฎเพื่อดำเนินงานสามารถใช้เป็นฮิวริสติกหรือเป็นความรู้พื้นฐาน (background knowledge) เพื่อใช้ช่วยในการจัดลำดับความสำคัญของกฎต่างๆ ในฐานความรู้

- Knowledge Validation and Reorganization Subsystem เป็นส่วนที่ทำหน้าที่ปรับปรุงฐานความรู้เพื่อจัดลำดับกฎให้เหมาะสมต่อการใช้งานมากขึ้น และทำหน้าที่ยืนยันความถูกต้องของความรู้ที่สังเคราะห์ขึ้นใหม่
- Explanation and Reasoning Facility เป็นโปรแกรมที่ช่วยเสริมการทำงานของระบบผู้เชี่ยวชาญให้สามารถอธิบายเหตุผลประกอบคำแนะนำที่แสดงแก่ผู้ใช้ระบบ
- User Interface เป็นส่วนติดต่อกับผู้ใช้เพื่อรับปัญหา และแสดงข้อแนะนำที่จะช่วยแก้ปัญหา



รูปที่ 3 โครงสร้างของระบบผู้เชี่ยวชาญเชิงอุปนัย

4. การสังเคราะห์กฎและจัดการกฎ

จากโครงสร้างในรูปที่ 3 ส่วนประกอบหลักที่ทำให้ระบบผู้เชี่ยวชาญเชิงอุปนัยแตกต่างจากระบบผู้เชี่ยวชาญทั่วไป คือ ส่วน Automatic Knowledge Acquisition Subsystem และส่วน Knowledge Validation and Reorganization Subsystem ส่วนประกอบแรกที่ทำหน้าที่ สังเคราะห์ความรู้ในรูปแบบของกฎสามารถใช้อัลกอริทึมการสังเคราะห์กฎแบบ greedy algorithm [11-14] แสดงขั้นตอนได้ดังต่อไปนี้

Algorithm Rule Induction

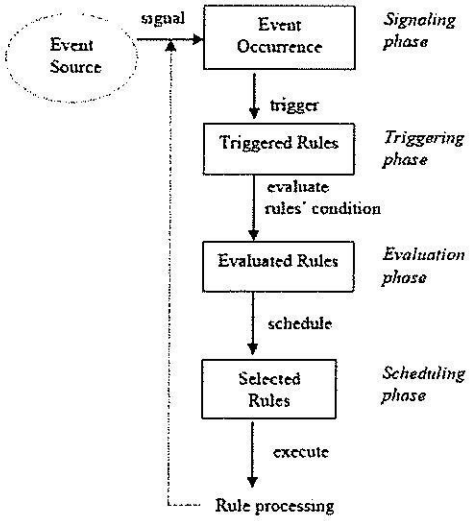
Input: Example cases

Output: Rule set

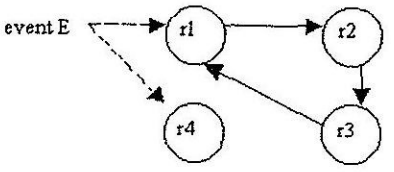
1. Create rule R from existing cases by greedily adding conditions that minimize error
2. Add R to the rule set RS
3. Remove cases covered by R and continue with step 1 until all cases are covered
4. Return RS

ในส่วนของการจัดการกับกฎทั้งหลายในฐานความรู้ เป็นการติดตามการเรียกใช้กฎในขณะระบบผู้เชี่ยวชาญทำงานเพื่อตรวจสอบว่ากฎใดบ้างที่ไม่ถูกใช้ประโยชน์ กฎที่ไม่ถูกใช้งานจะถูกทำเครื่องหมายไว้เพื่อการคัดทิ้งในภายหลัง โมเดลของการติดตามการเรียกใช้กฎใช้รูปแบบเดียวกับการทำงานของ ทริกเกอร์ (trigger) ในฐานข้อมูล แสดงโครงสร้างของโมเดลได้ดังรูปที่ 4

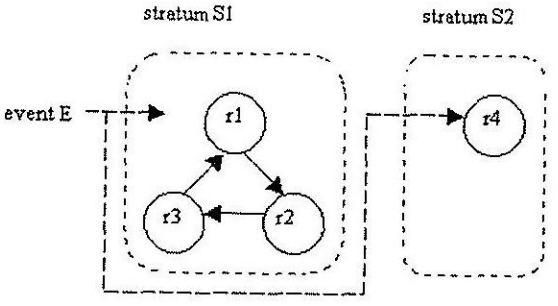
ในขั้นตอนการทำทริกเกอร์ (triggering phase) จะมีการตรวจสอบว่ามีกฎใดบ้างที่จะถูกเรียกใช้ ถ้ามีมากกว่าหนึ่งกฎ จะมีการจัดลำดับความสำคัญ และถ้าการเรียกใช้กฎมีลักษณะการทำงานต่อเนื่องไปยังกฎอื่นๆ จะติดตามการเรียกใช้นั้นในลักษณะของกราฟ ดังตัวอย่างในรูปที่ 5 ถ้ากราฟเกิดวัฏจักร (cycle) แสดงว่าการทำงานของระบบผู้เชี่ยวชาญอาจจะไม่สิ้นสุดสามารถแก้ไขปัญหานี้ได้ด้วนการแยกกลุ่มของกฎออกจากกันดังภาพในรูปที่ 6 และใช้อัลกอริทึม Rule Conflict Resolution ในการแก้ปัญหาความขัดแย้งของกฎ



รูปที่ 4 โมเดลการติดตามการทำงานของกฎในลักษณะของทริกเกอร์



รูปที่ 5 การเรียกใช้กฎในลักษณะกระตุ้นต่อเนื่องเป็นวัฏจักร



รูปที่ 6 การจัดโครงสร้างกฎโดยแยกออกเป็นชั้น

Algorithm 4.4 Trigger conflict resolution algorithm.

Input: an unordered stratum set S, a relational active database R and its metadata

Output: an ordered stratum set O in which the priority of each stratum has been assigned.

Steps:

6. $Active\_Rule\_Set_i \leftarrow Activate(S_i, E)$

   /* Activate every stratum S_i, $S_i \in$ S, such that the occurrence of an event E can invoke its trigger rule(s), and record all affected rules in the corresponding *Active_Rule_Set*. */

7. $K_i \leftarrow Induce(Active\_Rule\_Set_i)$

   /* The knowledge induction method (such as association-rule learning, decision-tree induction) is applied to induce knowledge from the content of each *Active_Rule_Set*. The induced knowledge is stored in K_i. */

8. $q_i = Degree\_of\_Constraint\ (K_i, metadata)$

   /* Calculate the value q, or the *Degree_of_Constraint*, of each set of induced knowledge K comparing to the integrity constraints given as a metadata. */

9. *sort(i, q)* /* Apply any sorting algorithm on the q-value associated with each stratum S_i. */

10. *return an ordered stratum set* $O = \{ S_i \mid S$ *has been sorted by its index* $i \}$

---

5. สรุป

บทความนี้นำเสนอความก้าวหน้าในโครงการวิจัยการออกแบบและพัฒนาระบบผู้เชี่ยวชาญเชิงอุปนัย ที่เน้นการแสดงความรู้ในรูปแบบกฎ ระบบผู้เชี่ยวชาญเชิงอุปนัยจัดเป็นระบบผู้เชี่ยวชาญในรุ่นที่สองซึ่งมีประเด็นแตกต่างจากในรุ่นแรกตรงที่การแสวงหาความรู้เพื่อสร้างฐานความรู้มีความเป็นอัตโนมัติมากขึ้น ความเป็นอัตโนมัติเกิดขึ้นจากการใช้เทคนิคการเรียนรู้แบบอุปนัย (inductive learning) ที่สามารถสังเคราะห์ความรู้ขึ้นจากกรณีตัวอย่างต่างๆ ที่ผ่านมาในอดีต งานวิจัยนี้นำเสนอกรอบแนวคิดที่เป็นแนวทางการออกแบบระบบผู้เชี่ยวชาญ และนำเสนอแนวทางการพัฒนาอัลกอริทึมเพื่อการสังเคราะห์ความรู้ในรูปแบบกฎ รวมถึงอัลกอริทึมการจัดการเพื่อคัดเลือกกฎและจัดลำดับความสำคัญของกฎ แนวทางการพัฒนางานวิจัยในขั้นตอนต่อไปจะเป็นการปรับปรุงการทำงานของเครื่องอนุมาน (inference engine) ให้ทำงานได้ดีขึ้นทั้งในเชิงนิรนัยและอุปนัย (deductive and inductive inference)

เอกสารอ้างอิง

[1] Giarratans, J.C. and Riley, G.D. (2005) *Expert systems: Principles and Programming*, fourth edition, Canada: Thomason Learning.

[2] Neale, I.M. (1988) First generation expert systems: A review of knowledge acquisition methodologies, The Knowledge Engineering Review, vol. 3, no. 2.

[3] Lavrac, N. and Mozetic, I (1992) Second generation knowledge acquisition methods and their application to medicine, In Keravanov, E.(Ed.) *Deep Models foo Medical Knowledge Engineering*, 177-198, Elsevier.

[4] Mookerjee, V.S.(2001) Debiasing traning data for inductive expert system construction, *IEEE Transactions on knowledge and Data Engineering*, vol. 13, no. 3, 497-512.

[5] Turney, p. (1995) Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm, *Journal of Artificial Intelligence Reserch*, vol. 2, 369-409.

[6] Irani, k., Cheng, J., Fayyad,U. and Qian,Z. (1993) Applying machine learning to semiconductor manufacturing, *IEEE Expert*, vol. 8, no.1, 41-47.

[7] Tam, k. and Kiang, M.(1990) Predicting bank failures: A neural network approach, *Applied Artificial Intelligence*, vol. 4, 265-280.

[8] Attar Software (2005) Knowledge builder for capturing, maintaining, deploying business rules in eBusiness systems [access on March 2005 form www.attar.com/White Papers/]

[9] Hart, D.(1986) *Knowledge Acquisition for Expert systems*, London: Kogan Page.

[10] Hart, D.(1985) The role of induction in knowledge elicitation, *Export Systems*, vol. 2, 24-28.

[11] Michalski, R. and Chilausky, c. (1980) Learning by being told and learning from examples: An experimental comparison of two methods of knowledge acquisition in the context of building an expert system for soybean disease diagnosis, *International Journal of Policy Analysis and Information Systems*, vol.4, 125-161.

[12] Weiss, S.M., Buckley, S.J., Kapoor, S. and Damgaard, S.(2003) Knowledge-based data mining, *Proceedings of SIGKDD*, August 24-27, 456-461.

[13] Cohen, W.(1995) Fast effective rule induction, *Proceesings of the Twelfth International Conference on Machine Learnings*, 115-123.

[14] Weiss, S. and Indurkhya, N.(1993) Optimized rule induction, *IEEE Expert*, vol. 8, no.6, 61-69.

ประวัติผู้วิจัย

รองศาสตราจารย์ ดร.นิตยา เกิดประสพ สำเร็จการศึกษาในระดับปริญญาเอกสาขา Computer Science จาก Nova Southeastern University เมือง Fort Lauderdale รัฐฟลอริดา ประเทศสหรัฐอเมริกา เมื่อปีพุทธศักราช 2542 (ค.ศ. 1999) ด้วยทุนการศึกษาของกระทรวงวิทยาศาสตร์ฯ โดยทำวิทยานิพนธ์ระดับปริญญาเอกในหัวข้อเรื่อง "The application of inductive logic programming to support semantic query optimization" หลังสำเร็จการศึกษาได้ปฏิบัติราชการในตำแหน่งอาจารย์ ประจำสาขาคอมพิวเตอร์ ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ต่อมาในปีพุทธศักราช 2543 ได้มาปฏิบัติงานในตำแหน่งอาจารย์ประจำสาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จนถึงปัจจุบัน งานวิจัยที่ทำในขณะนี้คือการพัฒนาระบบเหมืองข้อมูลประสิทธิภาพสูงที่สามารถทนต่อข้อมูลรบกวน และการเพิ่มความสามารถในการจัดการความรู้ของระบบเหมืองข้อมูล